Decision Tree: Example
Decision Tree / ID3 Pros

- Intuitive / explainable
- Can handle categorical and real-valued features
- Automatically performs feature selection
- The ID3 algorithm has a preference for shorter trees (simpler hypotheses)
• The ID3 algorithm is greedy (it selects the feature with the highest information gain at every step) so no optimality guarantee

• Overfitting
  • Can be addressed via heuristics ("regularization") or pruning ("validation").
Addressing Overfitting

- **Heuristics ("regularization"):**
  - Do not split leaves past a fixed depth $\delta$
  - Do not split leaves with fewer than $c$ labels
  - Do not split leaves where the maximal information gain is less than $\tau$
  - Predict the most common label at each leaf

- **Pruning ("validation"):**
  - Evaluate each split using a validation set
  - Compare the validation error with and without that split (replacing it with the most common label at that point)
The diagram and table illustrate a decision process with variables and outcomes.

### Table: $D_{val}$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>During</td>
<td>Backpack</td>
<td>Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Both</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>During</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
</tbody>
</table>
Pruning: 
Example

- Input: a decision tree, $t$ and a validation dataset, $D_{val}$
- Compute the validation error of $t$, $E_{val}(t)$
- For each split, $s \in t$
  - Compute $E_{val}(t \setminus s) = \text{the validation error of } t \text{ with } s \text{ replaced by a leaf using the most common label at } s$
- If $\exists$ a split $s \in t$ s.t. $E_{val}(t \setminus s) \leq E_{val}(t)$, repeat the pruning process with $t \setminus s^*$ where $t \setminus s^*$ is the pruned tree with minimal validation error (shorter trees win ties)
- Output: a pruned decision tree $t \setminus s^*$
\( D_{val} = \)

\( E_{val}(t) = 0.2 \)
$E_{val}(t) = 0.2$
\[
\mathcal{D}_{val} = \left\{ \begin{array}{c|c|c|c|c|c}
 x_1 & x_2 & x_3 & x_4 & y \\
 \hline
 \text{Rain} & \text{During} & \text{Backpack} & \text{Tired} & \text{Metro} \\
 \text{Rain} & \text{After} & \text{Both} & \text{Not Tired} & \text{Metro} \\
 \text{No Rain} & \text{Before} & \text{Backpack} & \text{Not Tired} & \text{Metro} \\
 \text{No Rain} & \text{During} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
 \text{No Rain} & \text{After} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
\end{array} \right. 
\]

\[
E_{val}(t | s_1) 
\]
\[ D_{val} = \]

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>During</td>
<td>Backpack</td>
<td>Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Both</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>During</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
</tbody>
</table>

\[ E_{val}(t|s_1) = 0.4 \]
$\mathcal{D}_{val} = \begin{cases} 
\text{Rain} & \text{During} & \text{Backpack} & \text{Tired} & \text{Metro} \\
\text{Rain} & \text{After} & \text{Both} & \text{Not Tired} & \text{Metro} \\
\text{No Rain} & \text{Before} & \text{Backpack} & \text{Not Tired} & \text{Metro} \\
\text{No Rain} & \text{During} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
\text{No Rain} & \text{After} & \text{Lunchbox} & \text{Tired} & \text{Drive} 
\end{cases}$

$E_{val}(t \setminus s_2) = 0.4$
\[
\mathcal{D}_{val} =
\begin{array}{cccccc}
\begin{array}{c|c|c|c|c|c|c}
\hline
x_1 & x_2 & x_3 & x_4 & y \\
\hline
\text{Rain} & \text{During} & \text{Backpack} & \text{Tired} & \text{Metro} \\
\text{Rain} & \text{After} & \text{Both} & \text{Not Tired} & \text{Metro} \\
\text{No Rain} & \text{Before} & \text{Backpack} & \text{Not Tired} & \text{Metro} \\
\text{No Rain} & \text{During} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
\text{No Rain} & \text{After} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
\hline
\end{array}
\end{array}
\]

\[
\begin{array}{cccccc}
\begin{array}{c|c|c|c|c|c}
\hline
s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\
\hline
0.4 & 0.4 & 0.4 & 0 & 0 & 0.2 \\
\hline
\end{array}
\end{array}
\]
\[
\mathcal{D}_{val} = \begin{array}{cccccc}
\text{x}_1 & \text{x}_2 & \text{x}_3 & \text{x}_4 & y \\
\text{Rain} & \text{During} & \text{Backpack} & \text{Tired} & \text{Metro} \\
\text{Rain} & \text{After} & \text{Both} & \text{Not Tired} & \text{Metro} \\
\text{No Rain} & \text{Before} & \text{Backpack} & \text{Not Tired} & \text{Metro} \\
\text{No Rain} & \text{During} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
\text{No Rain} & \text{After} & \text{Lunchbox} & \text{Tired} & \text{Drive} \\
\end{array}
\]
\[ D_{val} = \]

\[ E_{val}(t) = 0 \]
The ID3 algorithm is greedy (it selects the feature with the highest information gain at every step) so no optimality guarantee.

Overfitting
- Can be addressed via heuristics ("regularization") or pruning ("validation"): 

High variance
Bias-Variance Tradeoff (Example)
Bias-Variance Tradeoff (Example)

\[ \mathbb{E}_D [E_{out}(g_D)] = \mathbb{E}_{\tilde{x}} [\text{Variance of } g_D(\tilde{x}) + \text{Bias of } \bar{g}(\tilde{x})] \]
<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>Before</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>Rain</td>
<td>During</td>
<td>Both</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>During</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Backpack</td>
<td>Tired</td>
<td>Bike</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Lunchbox</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>During</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>During</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Bike</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Tired</td>
<td>Bike</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Both</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Lunchbox</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
</tbody>
</table>
## Data

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain</td>
<td>Before</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>Rain</td>
<td>During</td>
<td>Both</td>
<td>Not Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>Rain</td>
<td>During</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>Rain</td>
<td>After</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Backpack</td>
<td>Tired</td>
<td>Bike</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Lunchbox</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>Before</td>
<td>Lunchbox</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>During</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>During</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Not Tired</td>
<td>Bike</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Backpack</td>
<td>Tired</td>
<td>Bike</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Both</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Both</td>
<td>Tired</td>
<td>Drive</td>
</tr>
<tr>
<td>No Rain</td>
<td>After</td>
<td>Lunchbox</td>
<td>Not Tired</td>
<td>Metro</td>
</tr>
</tbody>
</table>
Decision Tree: Example
Bagging

- Short for Bootstrap aggregating

- Combines the prediction of many hypotheses to reduce variance

- If $n$ independent random variables $x_1, x_2, \ldots, x_n$ all have variance $\sigma^2$, then the variance of $\frac{1}{n} \sum_{i=1}^{n} x_i$ is $\frac{\sigma^2}{n}$
Bootstrapping

- A statistical method for estimating properties of a distribution, given (potentially a small number of) samples from that distribution
- Relies on resampling the samples \textit{with replacement} many, many times
Suppose you want to know the mean of a distribution so you draw 8 samples from that distribution: $D = \{1.70, -0.23, 0.54, -0.38, -1.53, 0.84, 0.60, 1.84\}$

Resample 8 values (with replacement) from $D$ 1000 times:

$\{-0.23, 0.54, -0.38, -1.53, -0.23, -0.38, -0.23, -1.53\}$

$\{-0.38, 0.60, -0.38, 1.85, -0.38, -0.38, 1.84, -0.23\}$

$\vdots$

$\{1.84, 0.84, 1.84, -1.53, 1.84, 1.84, 1.84, -0.23\}$
Bootstrapping: Example

- Suppose you want to know the mean of a distribution so you draw 8 samples from that distribution: $D = \{1.70, -0.23, 0.54, -0.38, -1.53, 0.84, 0.60, 1.84\}$

- Resample 8 values (with replacement) from $D$ 1000 times

- Compute the mean of each new resampled set

- Use these means to build point estimates (e.g. 0.43) or confidence intervals (e.g. $[-0.31, 1.12]$)
• Combining multiple hypotheses, \( \{h_1, h_2, \ldots, h_m\} \), to arrive at a single hypothesis

• Regression: average the predictions
  \[
  \bar{h}(\tilde{x}) = \frac{1}{m} \sum_{i=1}^{m} h_i(\tilde{x})
  \]

• Classification: find the category that the most hypotheses predict (plurality vote)
Bagging Decision Trees

- Input: \( D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, B \)

- For \( b = 1, 2, \ldots, B \)
  - Create a dataset, \( D_b \), by sampling \( n \) points from \( D \) with replacement
  - Learn a decision tree, \( t_b \), using \( D_b \) and the ID3 algorithm

- Output: \( \bar{t} \), the aggregated hypothesis
Bagging

- Short for **Bootstrap aggregating**

- Combines the prediction of many hypotheses to reduce variance

- If $n$ independent random variables $x_1, x_2, \ldots, x_n$ all have variance $\sigma^2$, then the variance of $\frac{1}{n} \sum_{i=1}^{n} x_i$ is $\frac{\sigma^2}{n}$
Split-Feature Randomization

- Predictions made by trees trained on similar datasets are highly correlated
- To decorrelate these predictions, randomly limit the features available at each iteration of the ID3 algorithm
- Every time the ID3 algorithm goes to split an impure leaf, randomly select $m < d$ features and only allow the algorithm to use one of those $m$ features.
  - For classification, a common choice is $m = \sqrt{d}$
  - For regression, a common choice is $m = \frac{d}{3}$
Random Forests

- **Input:** $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, B, m$

- **For** $b = 1, 2, \ldots, B$
  - Create a dataset, $\mathcal{D}_b$, by sampling $n$ points from $\mathcal{D}$ with replacement
  - Learn a decision tree, $t_b$, using $\mathcal{D}_b$ and the ID3 algorithm with **split-feature randomization**

- **Output:** $\bar{t}$, the aggregated hypothesis
Random Forests and Validation

• For each training point, $\bar{x}_i$, there are some trees which $\bar{x}_i$ was not used to train (roughly $B/e$); let these trees be $t_i^- = \{t_{i,1}, t_{i,2}, ..., t_{i,n_i}\}$

• Compute an aggregated prediction for each $\bar{x}_i$ using $t_i^-$:

\[
(\text{e.g. for regression}) \quad \bar{t}_i^- (\bar{x}_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} t_{i,j}^- (\bar{x}_i)
\]

• Compute the out-of-bag (OOB) error:

\[
E_{OOB} = \frac{1}{n} \sum_{i=1}^{n} e(y_i, \bar{t}_i^- (\bar{x}_i))
\]

• $E_{OOB}$ is almost an unbiased estimator of $E_{out}$
Random Forests and Feature Selection

- The interpretability of decision trees gets lost when we switch to random forests

- Random forests allow for the computation of “variable importance”, a way of ranking features based on how useful they are at predicting the output

- Initialize each feature’s importance to zero

- Each time a feature is chosen by the ID3 algorithm (with split-feature randomization), add that feature’s information gain (relative to the split) to its importance