Decision Tree

Implement Decision Tree algorithm as follows:

DTree(*records*, *attributes*) returns a tree → the best feature to split

If stopping criterion is met, return a leaf node with the assigned class.

Else pick an *attribute* F based on Gini Index and create a node *R* for it

For each possible value *v* of *F*:

Let *Sv* be the subset of records that have value *v* for *F*

call DTree(*Sv*, *attributes* – {*F*}) and attach the resulting tree as the subtree to the current node.

Return the subtree.

Example

- Golf dataset
  - 4 features
  - Label: yes/no
- We use multi-way
  Split in this assignment

Outlook	Temperature	Humidity	Windy	Label
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Stopping criteria

- stopCriteria(dataset)

```
        assignedLabel = None
        if all class labels are the same
                assignedLabel = label
        else if no more features to split
                assignedLabel = majority(labels)
```

Check if the data matrix has only one column left by evaluating the number of columns in the current dataset

- Input: dataset/split dataset
- Output: assigned label

- The original dataset does not satisfy the stopping criteria.
- Then we find best split feature:

Find best split feature

- chooseBestFeature(dataset)

  for each feature i in the dataset

  calculate gini index on dataset

  for each $value$ of the feature

  subset = splitData(dataset, i, $value$)

  calculate gini index on the subset

  calculate Gain for feature i

  Find the bestGain and the corresponding feature id

- Input: dataset/split dataset
- Output: index of best feature

Find best split feature

- Find best split feature
  - For each feature, calculate the gain of gini indexes

If Feature = Outlook

Number of "No" in the last column

$$\text{Gini}=1-\left(\frac{\boxed{5}}{\boxed{14}}\right)^2-\left(\frac{9}{14}\right)^2=0.46$$

Number of rows

Outlook = Rainy

$$\text{gini}=1-\left(\frac{\boxed{3}}{\boxed{5}}\right)^2-\left(\frac{2}{5}\right)^2=0.48$$

Number of "No" in the last column given Outlook = rainy

Number of rows given Outlook = rainy

Outlook	Temperature	Humidity	Windy	Label
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Find best split feature

- Find best split feature
  - For each feature, calculate the gain of gini indexes

If Feature = Outlook

$$\text{Gini}=1-\left(\frac{5}{14}\right)^2-\left(\frac{9}{14}\right)^2=0.46$$

Outlook = Rainy

$$\text{gini}=1-\left(\frac{3}{5}\right)^2-\left(\frac{2}{5}\right)^2=0.48$$

Outlook = overcast

$$\text{gini}=1-\left(\frac{4}{4}\right)^2=0$$

Outlook	Temperature	Humidity	Windy	Label
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Find best split feature

- Find best split feature
  - For each feature, calculate the gain of gini

If Feature = Outlook

$$\text{Gini}=1-\left(\frac{5}{14}\right)^2-\left(\frac{9}{14}\right)^2=0.46$$

Outlook = Rainy

$$\text{gini}=1-\left(\frac{3}{5}\right)^2-\left(\frac{2}{5}\right)^2=0.48$$

Outlook = overcast

$$\text{gini}=1-\left(\frac{4}{4}\right)^2=0$$

Outlook = sunny

$$\text{gini}=1-\left(\frac{3}{5}\right)^2-\left(\frac{2}{5}\right)^2=0.48$$

$$\text{Gain}=0.46-\left(\frac{5}{14}*0.48+\frac{4}{14}*0+\frac{5}{14}*0.48\right)=0.117$$

Outlook	Temperature	Humidity	Windy	Label
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No

Find best split feature

- Find best split feature
  - For each feature, calculate the gain
    - If Feature = Temperature, follow the same procedure to obtain the gain value
    - After the calculation for each feature on the dataset, we obtain

Gain(outlook) = 0.117

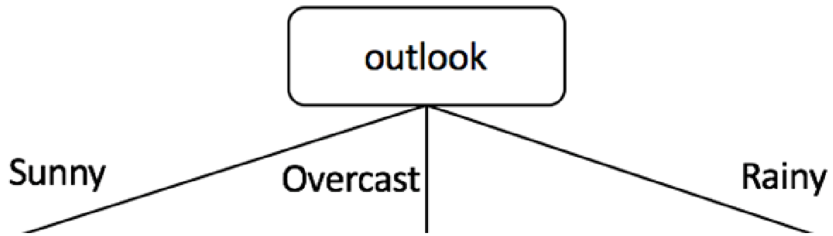
Gain(temperature)=0.018

Gain(humidity)=0.092

Gain(windy)=0.031

So Outlook is the best feature to split;

Then we split the data set:

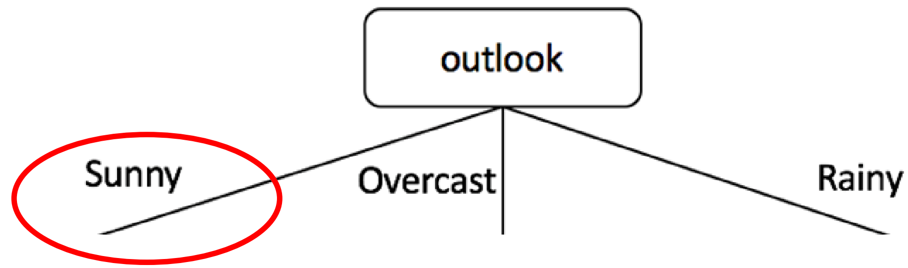


Split the dataset

- For each value in Outlook, split the dataset
  - splitData function has been provided.

Delete the Outlook column

Outlook	Temperature	Humidity	Windy	Label
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No



Temperature	Humidity	Windy	Label
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	Normal	FALSE	Yes
Mild	High	TRUE	No

Choose subset: Outlook = sunny;

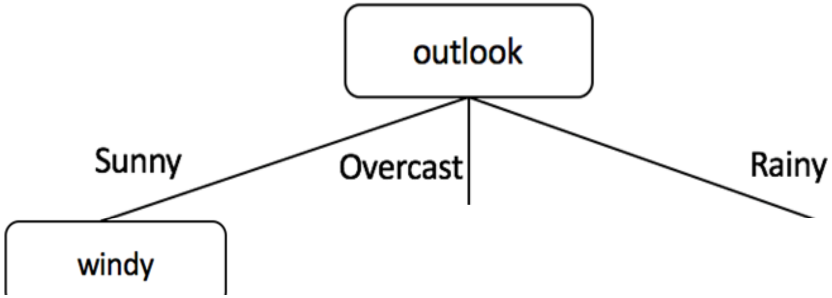
Repeat on sub-dataset

- Repeat the same procedure for the sub-dataset
  - This is done by calling Dtree again for each value of the best split feature (Outlook in our example).
  - It is already implemented in the template

- Sub-dataset doesn't satisfy stopping criteria
- Find best split on sub-dataset

(outlook=sunny)
Gain(temperature)=0.0133
Gain(Humidity)=0.0133
Gain(windy)=0.48
So windy is the best feature to split the sub-dataset (when outlook='sunny')

Temperature	Humidity	Windy	Label
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	Normal	FALSE	Yes
Mild	High	TRUE	No



Repeat on sub-dataset

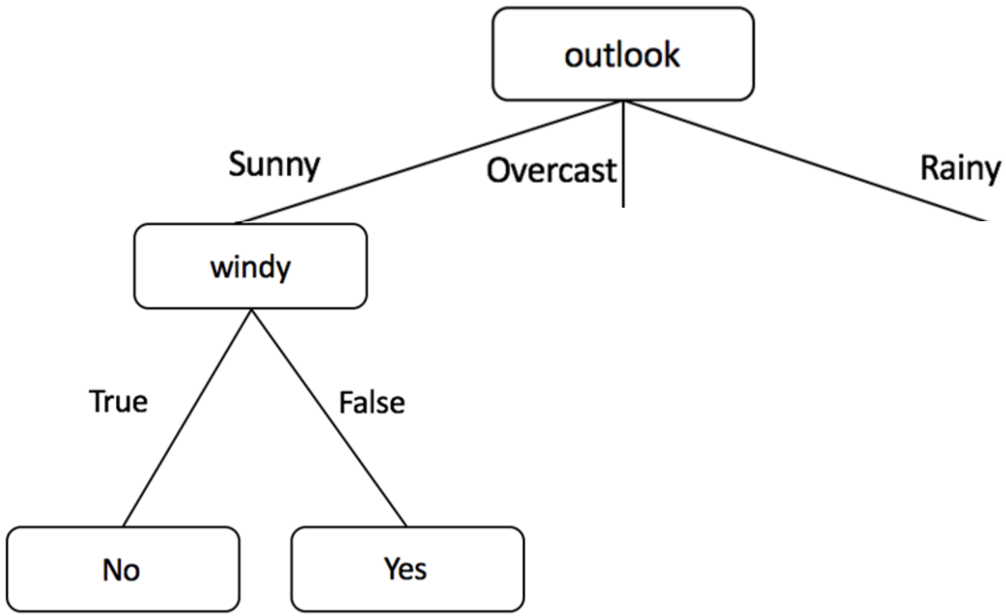
- Iterate until stopping criteria satisfied

Windy = False → Yes

Windy = True → No

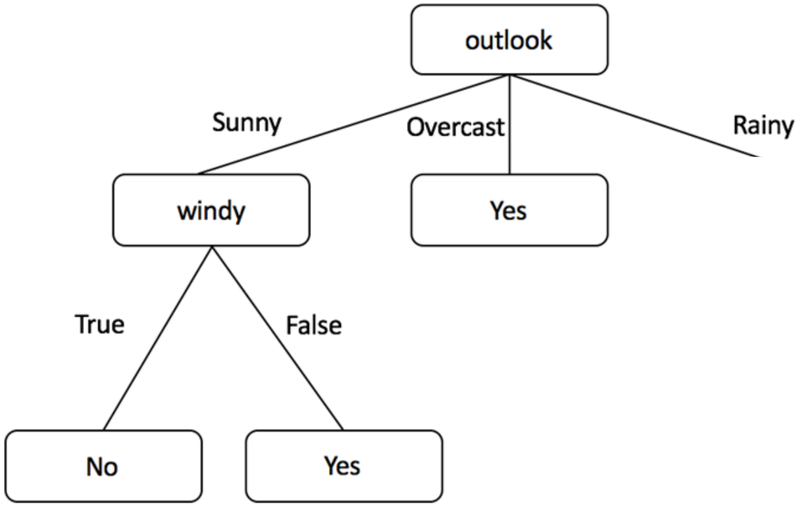
(outlook=sunny)

Temperature	Humidity	Windy	Label
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	Normal	FALSE	Yes
Mild	High	TRUE	No



Repeat on sub-dataset

- Follow the same procedure for overcast and Rainy
  - Outlook = overcast → label=yes

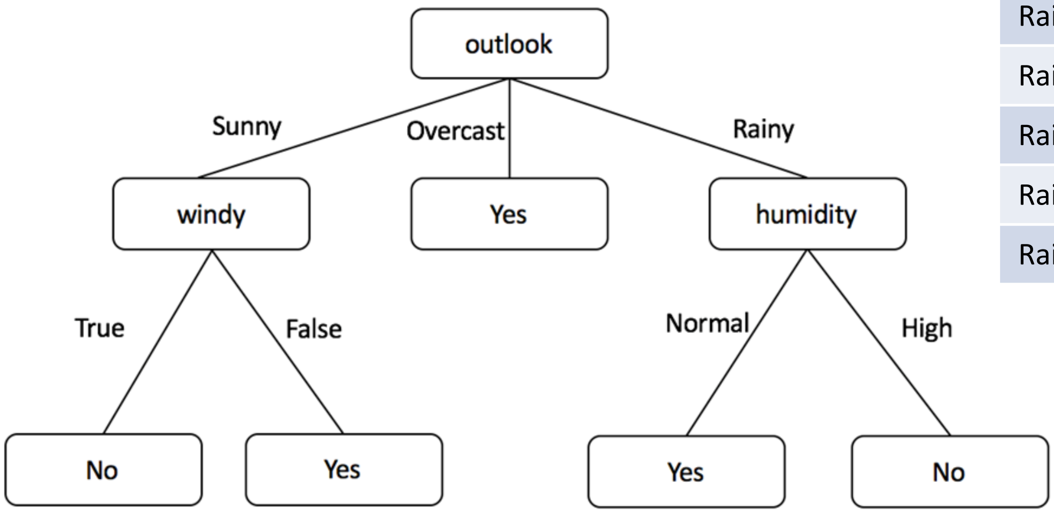


(outlook=Overcast)

Outlook	Temperature	Humidity	Windy	Label
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

Repeat on sub-dataset

- Follow the same procedure for overcast and Rainy
  - Outlook = Rainy



(outlook=Rainy)

Outlook	Temperature	Humidity	Windy	Label
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

Gain(temperature)=0.28
Gain(Humidity)=0.48
Gain(windy)=0.013

Humidity = High → No
Humidity = Normal → Yes