Decision Tree

DigiPen INSTITUTE OF TECHNOLOGY SINGAPORE

Recap: what is classification?

- Given a collection of records (*training dataset*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* that maps the relationship between the class and the other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test dataset* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to construct the model and test set used to validate it.

Tid	Refund	Marital Status	Taxable Income	Tax Evade
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

Advantages & Disadvantages of Classification Tree

- **Advantages:**
  - Provides visually intuitive output (Tree)
  - Simple to understand and interpret.
  - Data Requires little preparation. Outlier treatment is not needed

- **Disadvantages**
  - Classification tree models create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

Main issues of classification tree learning

- **Choosing the splitting criterion**
  - Impurity based criteria
  - Information gain
  - Statistical measures of association

- **Binary or multiway splits**
  - Multiway split
  - Binary split

- **Finding the right sized tree**
  - Stopping Criteria (Pre-pruning)
  - Post-pruning

Tree algorithms : ID3, C4.5, C5.0, CHAID and CART

- **CHAID – CHI-squared Automatic Interaction Detector.** The “Chi-squared” part of the name arises because the technique essentially involves automatically constructing many cross-tabs, and working out statistical significance of the proportions. The most significant relationships are used to control the structure of a tree diagram
  - CHAID is a non-binary decision tree; Recursive Partitioning Algorithm
  - Continuous variables must be grouped into a finite number of bins to create categories.

- **CLASSIFICATION AND REGRESSION TREES (CART)** are binary decision trees, which split a single variable at each node.
  - The CART algorithm recursively goes though an exhaustive search of all variables and split values to find the optimal splitting rule for each node.

- **ID3, C4.5, C5.0** builds decision trees from a set of training data using the concept of information entropy

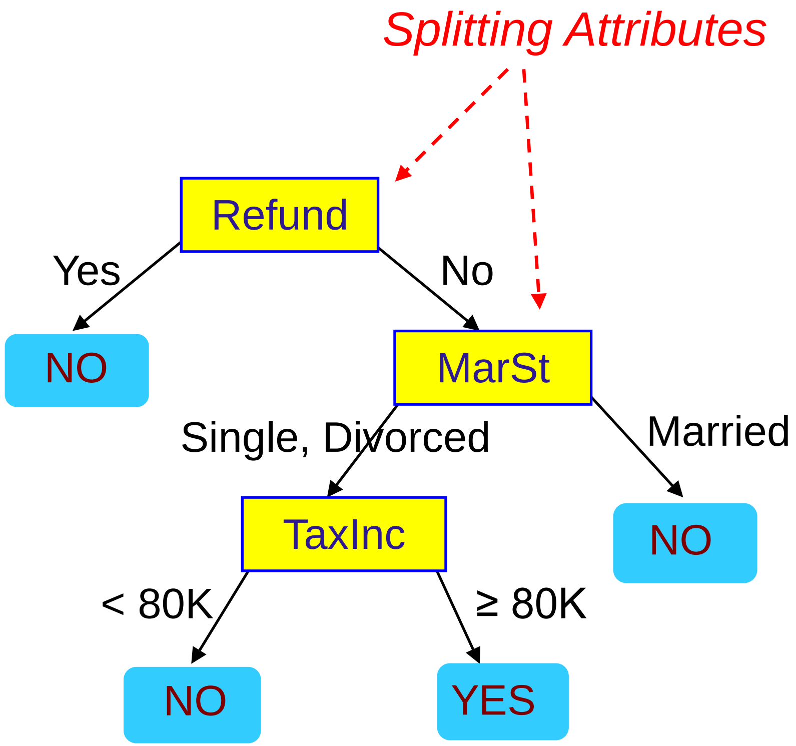
Decision Tree Example

- Example: use training data to build a decision tree model

Tid	Refund	Marital Status	Taxable Income	Tax Evade
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

Training Data

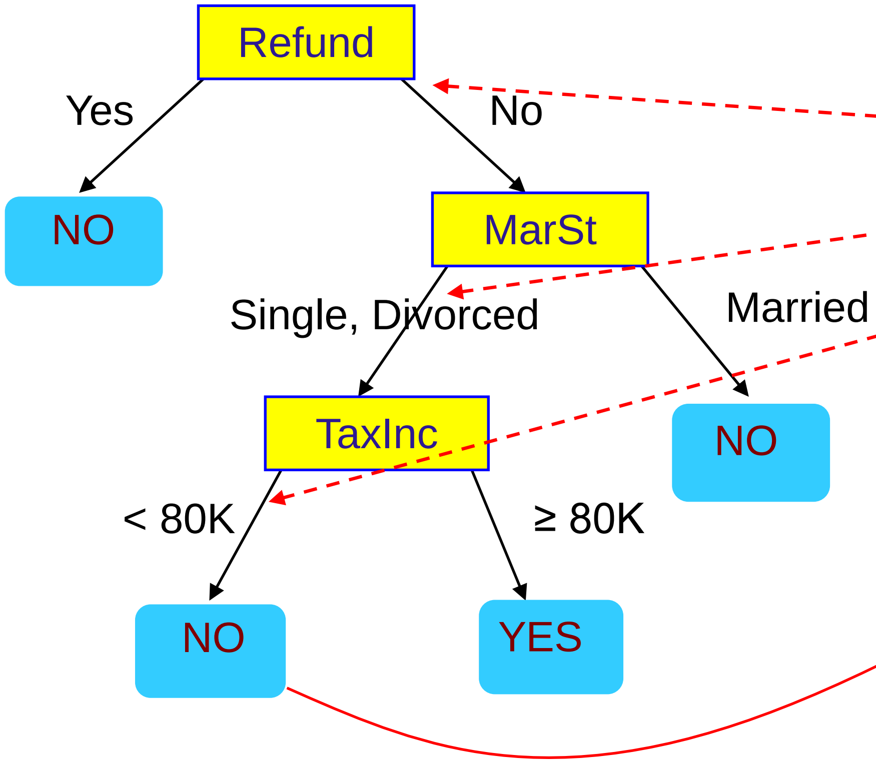
To build



Model: Decision Tree

Decision Tree Example

- Example: apply the trained decision tree model to make prediction for previously unseen data

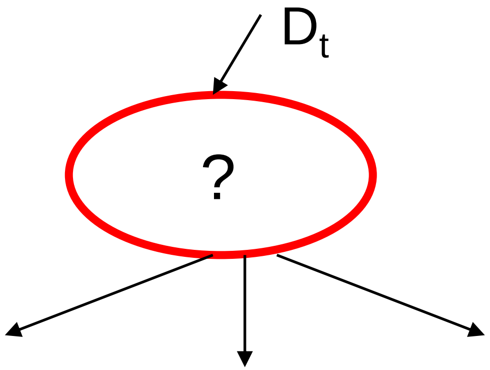


Test data (previously unseen)

Refund	Marital Status	Taxable Income	Tax Evade
No	Single	70K	?

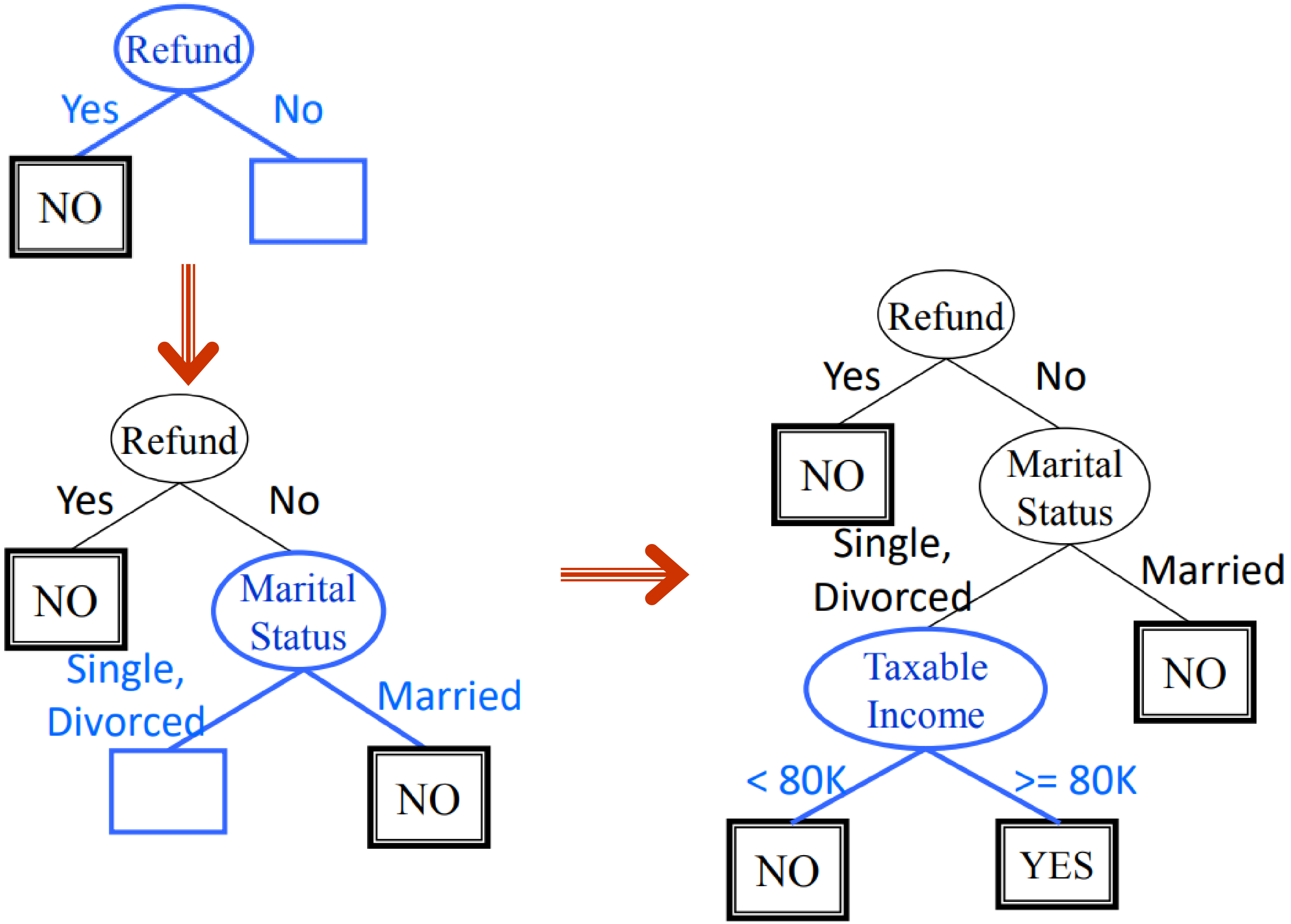
Decision Tree: Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- General Procedure:
  - If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
  - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
  - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset after splitting.



DigiPen INSTITUTE OF TECHNOLOGY SINGAPORE

Decision Tree: Hunt's Algorithm



Tid	Refund	Marital Status	Taxable Income	Tax Evade
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

How to Determine the Best Split from a Node

- After splitting, nodes with homogeneous class distribution are preferred

- So, need a measure of node impurity:

C0: 5 C1: 5	C0: 10 C1: 0
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

C0 is class 0 and C1 is class 1

Measures of Node Impurity

- Gini Index
- Entropy

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
- Minimum (0.0) when all records belong to one class, implying most interesting information

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

Splitting Based on GINI

- Used in CART (Classification and Regression Trees) algorithm by default. scikit-learn uses CART to implement its decision tree.
- When a parent node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child node i,

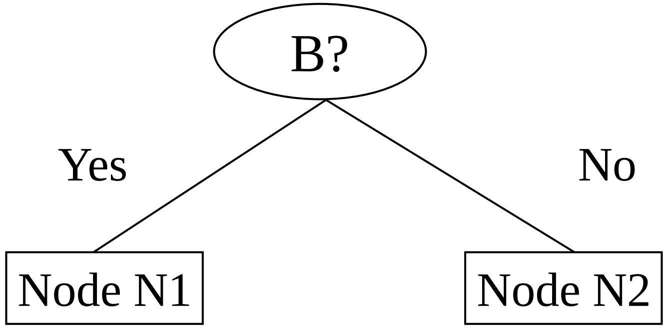
n = number of records at parent node p.

DigiPen INSTITUTE OF TECHNOLOGY SINGAPORE

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
  - Larger and Purer Partitions are sought for.

	Parent
C1	**7**
C2	**5**
Gini = 0.486	



Gini(N1)
$= 1 – (5/6)^2 – (1/6)^2$
$= 0.278$

Gini(N2)
$= 1 – (2/6)^2 – (4/6)^2$
$= 0.444$

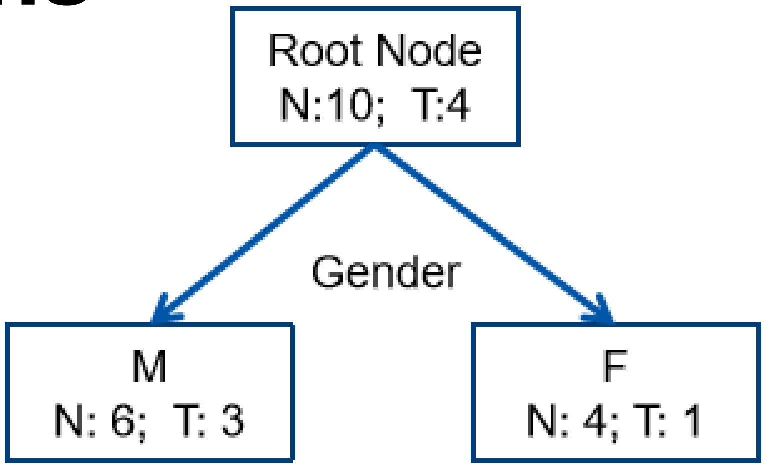
	N1	**N2**
C1	**5**	**2**
C2	**1**	**4**
Gini=0.361		

Weighted Gini of N1 N2
$= 6/12 * 0.278 +$
$6/12 * 0.444$
$= 0.361$

Gain = 0.486 – 0.361 = 0.125

Example: Gini Calculations

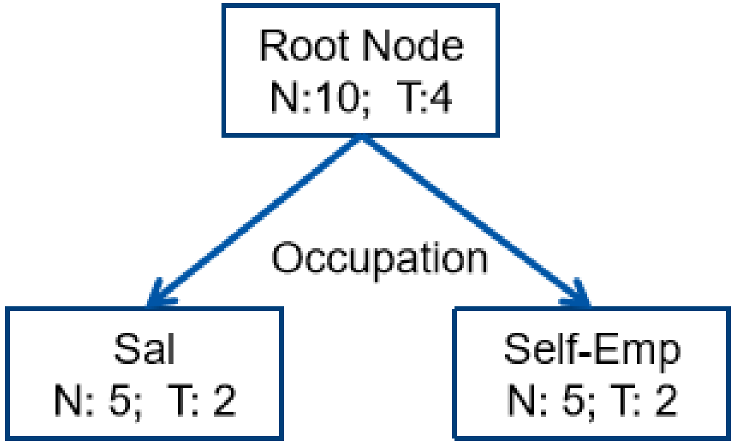
Cust_ID	Gender	Occupation	Age	Target
1	M	Sal	22	1
2	M	Sal	22	0
3	M	Self-Emp	23	1
4	M	Self-Emp	23	0
5	M	Self-Emp	24	1
6	M	Self-Emp	24	0
7	F	Sal	25	1
8	F	Sal	25	0
9	F	Sal	26	0
10	F	Self-Emp	26	0



Node	Gini Computation Formula	Gini Index
Overall	= 1 - ((4/10)^2 + (6/10)^2)	0.48
Gender = M	= 1 - ((3/6)^2 + (3/6)^2)	0.50
Gender = F	= 1 - ((1/4)^2 + (3/4)^2)	0.375
Gender	= (6/10) * 0.5 + (4/10) * 0.375	0.45
Gini Gain	= Gini (Overall) – Gini (Gender)	0.03

DIGIPEN INSTITUTE OF TECHNOLOGY SINGAPORE

Gini calculations

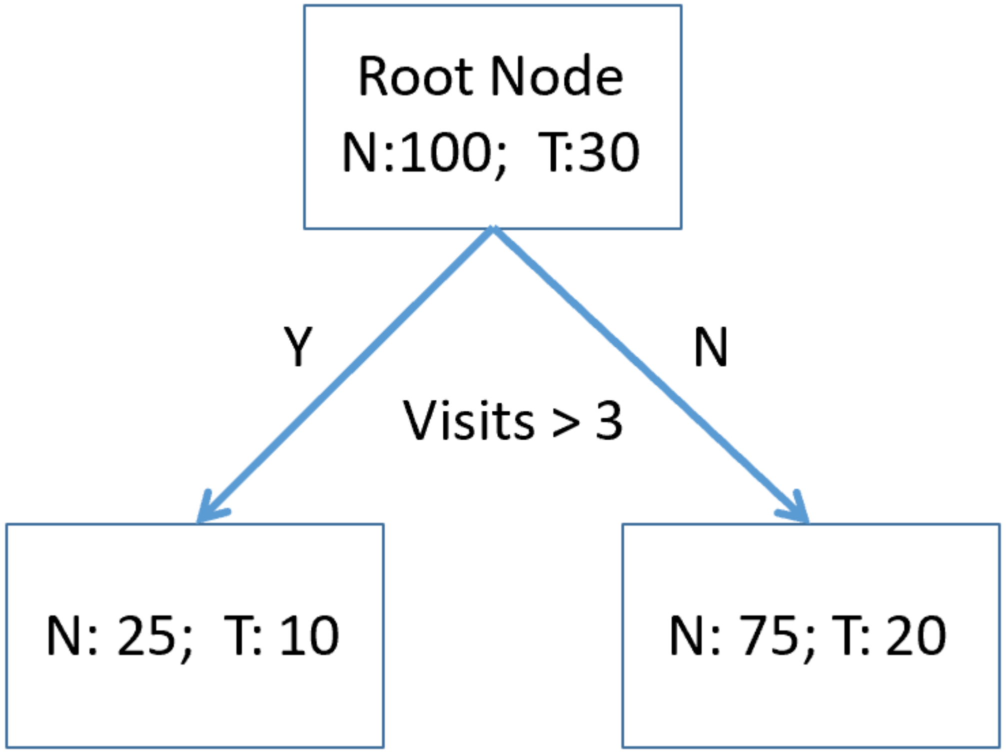


Node	Gini Computation Formula	Gini Index
Overall	= 1 - ((4/10)^2 + (6/10)^2)	0.48
Occ = Sal	= 1 - ((2/5)^2 + (3/5)^2)	0.48
Occ = Self-Emp	= 1 - ((2/5)^2 + (3/5)^2)	0.48
Occupation	= (5/10) * 0.48 + (5/10) * 0.48	0.48
Gini Gain	= Gini (Overall) – Gini (Occupation)	**0.0**

Age	<=22	<=23	<=24	<=25
Gini (Left)	0.5	0.5	0.5	0.5
Gini (Right)	0.47	0.44	0.38	0
Gini Split	0.48	0.47	0.45	0.40
Gini Gain	0.0	0.01	0.03	0.08

DIGIPEN INSTITUTE OF TECHNOLOGY SINGAPORE

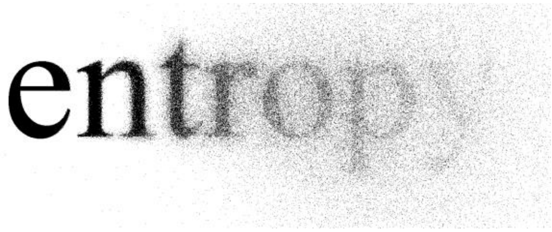
Exercise… Compute Gini Gain



DigiPen INSTITUTE OF TECHNOLOGY SINGAPORE

Measure of Impurity: Entropy

- Entropy at a given node t:



$$Entropy(t) = -\sum_{j} p(j\,|\,t) \log p(j\,|\,t)$$

base-2 log here, by convention

(NOTE: $p(j\,|\,t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
  - Maximum ($\log n_c$) when records are equally distributed among all classes implying least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

Information Gain based on Entropy

- Information Gain based on Entropy:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions (children);

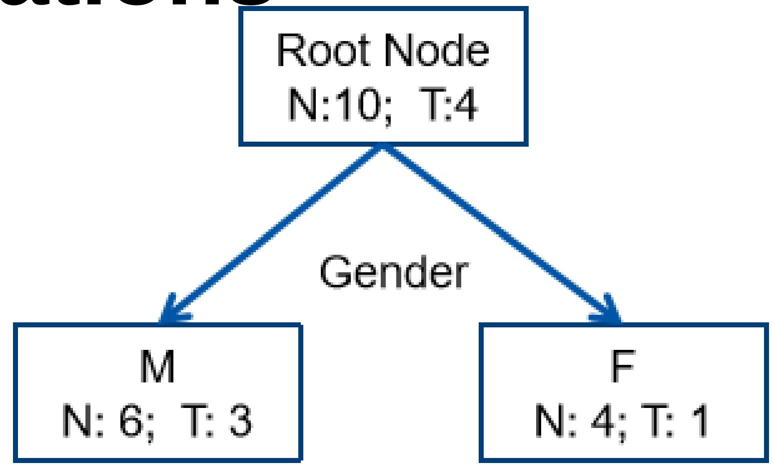
n_i is number of records in partition i

  - Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)

  - Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

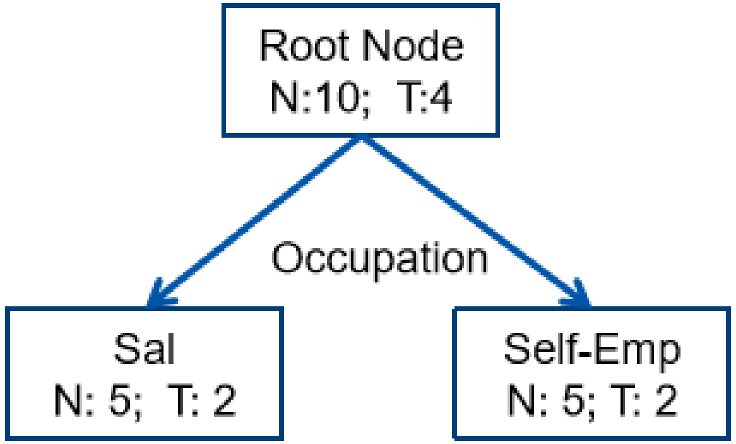
Example: Entropy Calculations

Cust_ID	Gender	Occupation	Age	Target
1	M	Sal	22	1
2	M	Sal	22	0
3	M	Self-Emp	23	1
4	M	Self-Emp	23	0
5	M	Self-Emp	24	1
6	M	Self-Emp	24	0
7	F	Sal	25	1
8	F	Sal	25	0
9	F	Sal	26	0
10	F	Self-Emp	26	0



Node	Entropy Computation Formula	Gini Index
Overall Entropy	$= -((4/10)\log_2(4/10) + (6/10)\log_2(6/10))$	0.971
Entropy of Gender = M	$= -((3/6)\log_2(3/6) + (3/6)\log_2(3/6))$	1
Entropy of Gender = F	$= -((1/4)\log_2(1/4) + (3/4)\log_2(3/4))$	0.811
Entropy of Gender Split	$= (6/10) * 1 + (4/10) * 0.811$	0.924
Information Gain	= Gini (Overall) – Gini (Gender)	0.047

Entropy calculations

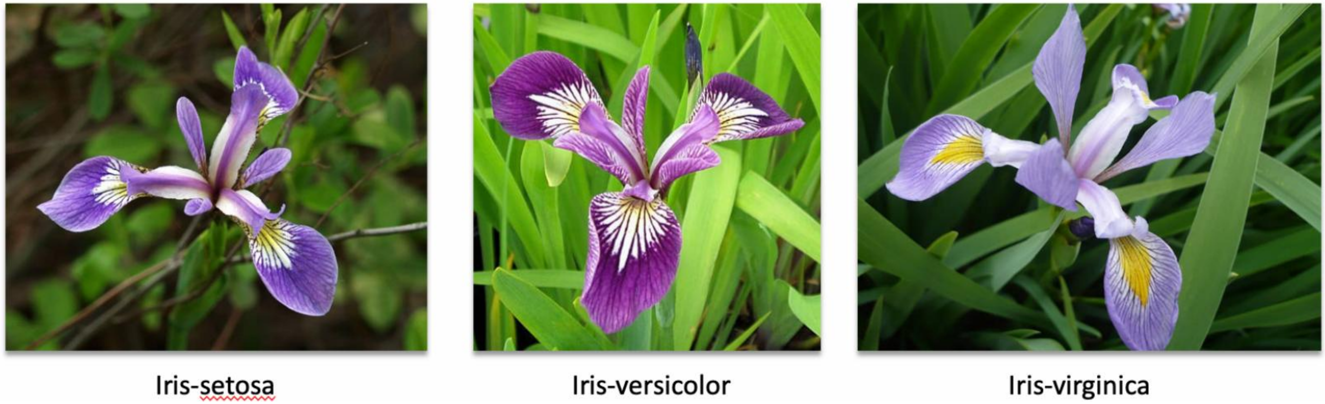


Node	Gini Computation Formula	Gini Index
Overall Entropy	$= -((4/10)\log_2(4/10) + (6/10)\log_2(6/10))$	0.97
Occ = Sal	$= -((2/5)\log_2(2/5) + (3/5)\log_2(3/5))$	0.97
Occ = Self-Emp	$= -((2/5)\log_2(2/5) + (3/5)\log_2(3/5))$	0.97
Entropy of Occupation Split	= (5/10) * 0.97 + (5/10) * 0.97	0.97
Information Gain	= Gini (Overall) – Gini (Occupation)	**0.0**

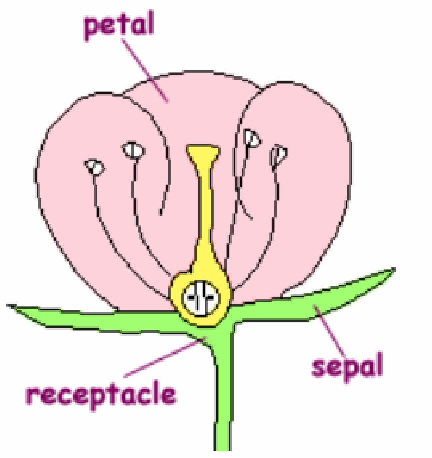
Age	<=22	<=23	<=24	<=25
Entropy (Left)	1.0	1.0	1.0	1.0
Entropy (Right)	0.955	0.918	0.811	0.0
Entropy Split	0.964	0.951	0.924	0.8
Information Gain	0.006	0.019	0.046	0.17

Application: Decision Tree for Iris Dataset

- Iris Flower dataset contains data of three species of Iris Flower:
  - Setosa, Versicolor, Virginica
- Number of records: 150
  - 50 records for each species
- Four features for each record:
  - Sepal Length, Sepal Width, Petal Length, Petal Width



- Based on the combination of these 4 features, build a decision tree classifier.



DigiPen INSTITUTE OF TECHNOLOGY SINGAPORE

Application: Decision Tree for Iris Dataset

[1]:
```python
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn import datasets
import matplotlib.pyplot as plt
```

[2]:
```python
iris = datasets.load_iris()
X = iris.data
y = iris.target

print("feature_names:\t", iris.feature_names)
print("target_names:\t", iris.target_names) # 0 for setosa, 1 for versicolor, 2 for virginica
```

```
feature_names:   ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
target_names:    ['setosa' 'versicolor' 'virginica']
```

Application: Decision Tree for Iris Dataset

[1]:
```python
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn import datasets
import matplotlib.pyplot as plt
```

[2]:
```python
iris = datasets.load_iris()
X = iris.data
y = iris.target

print("feature_names:\t", iris.feature_names)
print("target_names:\t", iris.target_names) # 0 for setosa, 1 for versicolor, 2 for virginica
```

```
feature_names:   ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
target_names:    ['setosa' 'versicolor' 'virginica']
```

Application: Decision Tree for Iris Dataset

[3]:
```
combined_X_y = np.concatenate((X, y.reshape(-1,1)), axis=1)
```

[4]:
```
iris_df = pd.DataFrame(combined_X_y, columns=['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)', 'class'])
iris_df
```

[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	class
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2.0
146	6.3	2.5	5.0	1.9	2.0
147	6.5	3.0	5.2	2.0	2.0
148	6.2	3.4	5.4	2.3	2.0
149	5.9	3.0	5.1	1.8	2.0

150 rows × 5 columns

[6]:
```
iris_df.groupby('class').size()
```

[6]:
```
class
0.0    50
1.0    50
2.0    50
dtype: int64
```

Application: Decision Tree for Iris Dataset

```
[7]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=7)
```

```
[8]: np.unique(y_train, return_counts=True)
```

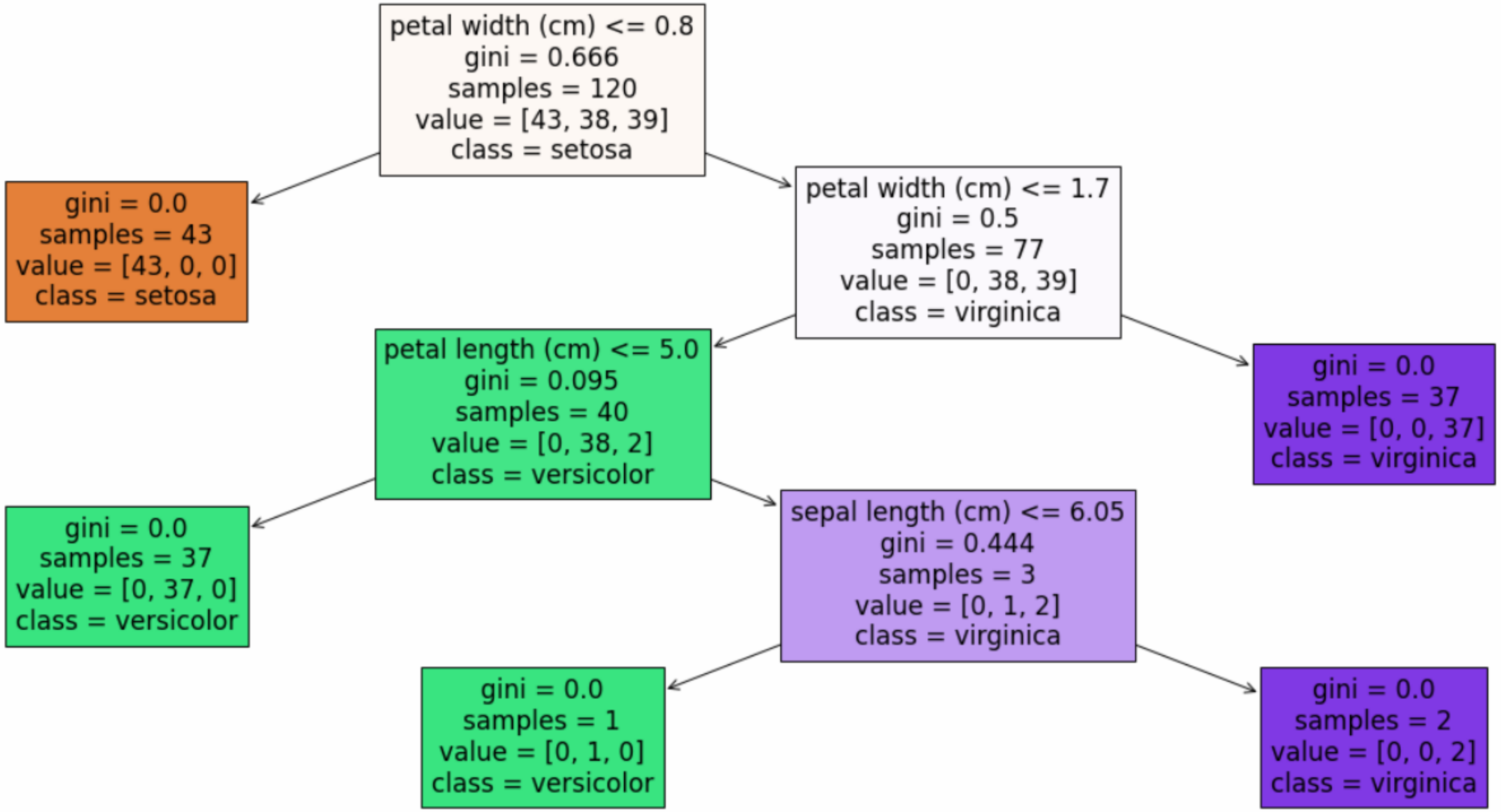
```
[8]: (array([0, 1, 2]), array([43, 38, 39], dtype=int64))
```

```
[9]: dtc = DecisionTreeClassifier() # default criterion is 'gini',
     dtc.fit(X_train, y_train) # train a decision tree model
```

```
[9]: ▾ DecisionTreeClassifier
     DecisionTreeClassifier()
```

Application: Decision Tree for Iris Dataset

```python
from sklearn import tree
fig = plt.figure(figsize=(25,10))
_ = tree.plot_tree(dtc, feature_names=iris.feature_names,
                   class_names=iris.target_names,
                   filled=True)
```



Application: Decision Tree for Iris Dataset

```python
y_test_predict = dtc.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_test_predict), "\n")

print("Confusion matrix:")
print(confusion_matrix(y_test, y_test_predict))

print("Classification report:")
print(classification_report(y_test, y_test_predict))
```

```
Accuracy: 0.9

Confusion matrix:
[[ 7  0  0]
 [ 0 10  2]
 [ 0  1 10]]
Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         7
           1       0.91      0.83      0.87        12
           2       0.83      0.91      0.87        11

    accuracy                           0.90        30
   macro avg       0.91      0.91      0.91        30
weighted avg       0.90      0.90      0.90        30
```


DIGIPEN INSTITUTE OF TECHNOLOGY SINGAPORE

Decision Tree Classifier Hyperparameters (not full)

- criterion : The function to measure the quality of a split.
  - “gini” for Gini Impurity
  - “entropy” for Information gain
- max_depth : The maximum depth of the tree.
- min_samples_split : The minimum number of samples required to split an internal node; the default is 2.
- min_samples_leaf : The minimum number of samples required to be a leaf node; the default is 1.
- max_features : The number of features to consider when looking for the best split

References

- Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2017). *Introduction to Data Mining* (2nd ed.). Pearson. https://www-users.cse.umn.edu/~kumar001/dmbook/index.php

- *Decision Trees*. Scikit-Learn. https://scikit-learn.org/stable/modules/tree.html

DigiPen

INSTITUTE OF TECHNOLOGY
SINGAPORE