Random Forest

REVIEW

Random Forest Classifier

- Consider the following dataset.

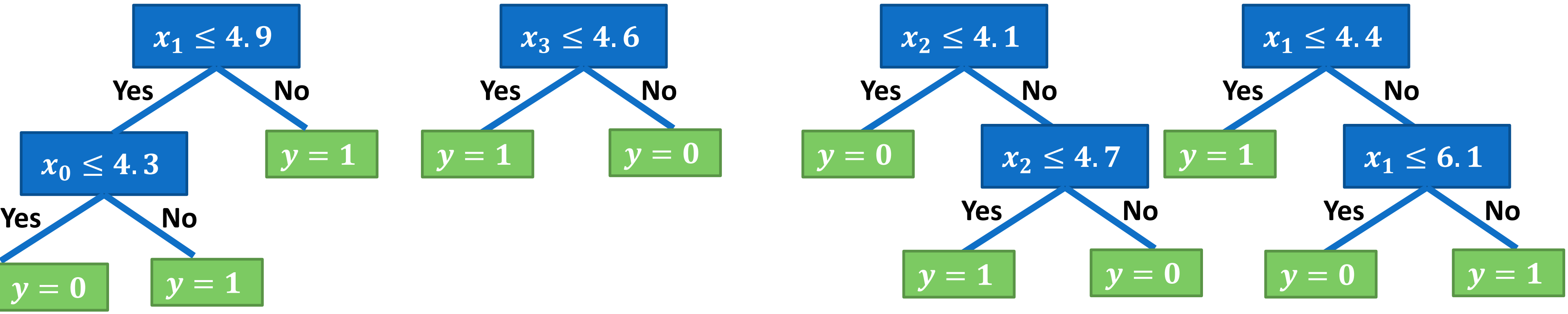
Step 3: Build a tree for each bootstrapped dataset

ID	x_0	x_1	y
2	2.7	4.8	0
0	4.3	4.9	0
2	2.7	4.8	0
4	6.5	2.9	1
5	2.7	6.7	1
5	2.7	6.7	1

ID	x_2	x_3	y
2	4.1	5.0	0
1	5.9	5.5	0
3	4.5	3.9	1
1	5.9	5.5	0
4	4.7	4.6	1
4	4.7	4.6	1

ID	x_2	x_4	y
4	4.7	6.1	1
1	5.9	5.9	0
3	4.5	5.9	1
0	4.1	5.5	0
0	4.1	5.5	0
2	4.1	5.6	0

ID	x_1	x_3	y
3	4.4	3.9	1
3	4.4	3.9	1
2	4.8	5.0	0
5	6.7	5.3	1
1	6.1	5.5	0
2	4.8	5.0	0

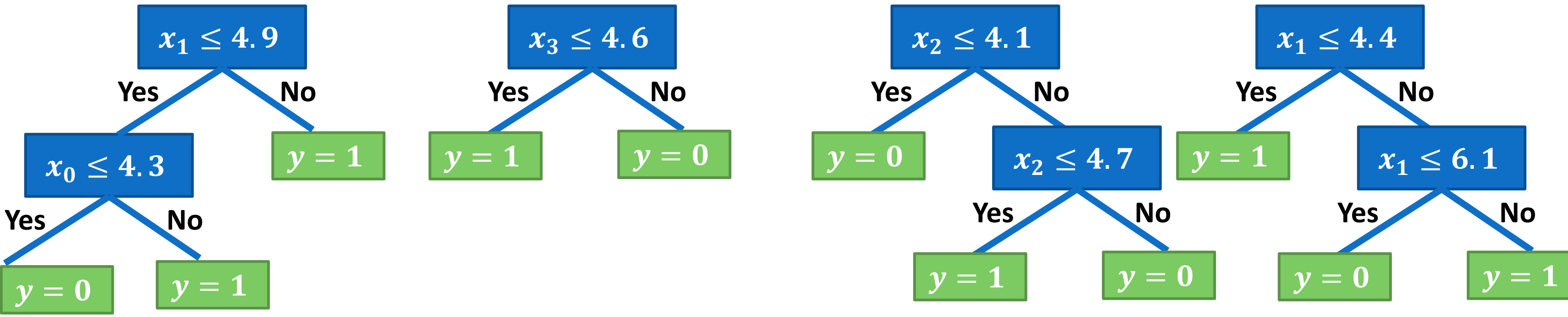


Note: A feature can repeat in a tree if it still has distinct values left (for that path).

Random Forest Classifier

- Consider following test record

Step 4: Use all trees for predictions…



$y = 1$ $y = 0$ $y = 1$ $y = 1$

x_0	x_1	x_2	x_3	x_4	y
2.8	6.2	4.3	5.3	5.5	?

Step 5: Consider majority vote to label the test record (Aggregating)

Bootstrapping + Aggregating is called "Bagging"

Note: Bagging can be done using any classifier (not just decision trees)

DT and RF for Regression

- **How to use decision trees and random forest for regression problems?**
- We can use minimum count of records at leaf node and then assign average as the label y
- Or
- We can compute standard deviation and standard deviation reduction (just like reduction in entropy).
  - The attribute with largest standard deviation reduction is chosen for decision node
  - we need some Terminate when:
    - Coefficient of deviation for a branch becomes smaller than a certain threshold (e.g., 10%) and/or when too few instances (n) remain in the branch (e.g., 3).
- **How good is the tree?**
  - Calculate training and testing errors

Self Study: https://www.saedsayad.com/decision_tree_reg.htm

Evaluation Metrics for Classification

How to evaluate the performance of the predictions?

- One straight forward method is to calculate accuracy on **testing split**

Actual Label	Pred. Label
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

Two mistakes out of 10

$$\frac{8}{10} = 0.8$$

$$0.8 \times 100 = 80\%$$

What's wrong with this kind of evaluation?

Not robust when label distribution is skewed

Does not tell a lot about the predictions…

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Positive (1)	Negative (0)
Positive (1)		
Negative (0)		

Note: Labels could be any binary class labels, such as **spam, not spam**. Its not necessarily have to be **negative, positive**

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions		Spam (1)	Not Spam (0)
	Spam (1)		
	Not Spam (0)		

Note: Labels could be any binary class labels, such as **spam, not spam**. Its not necessarily have to be **negative, positive**

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP**	
Not Spam (0)		

TP: We predicted “spam” and it was actually “spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP**	**FP**
Not Spam (0)		

TP: We predicted “spam” and it was actually “spam”

FP: We predicted “spam” and it was actually “not spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP**	**FP**
Not Spam (0)	**FN**	

TP: We predicted “spam” and it was actually “spam”

FP: We predicted “spam” and it was actually “not spam”

FN: We predicted “not spam” and it was actually “spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP**	**FP**
Not Spam (0)	**FN**	**TN**

TP: We predicted “spam” and it was actually “spam”

FP: We predicted “spam” and it was actually “not spam”

FN: We predicted “not spam” and it was actually “spam”

FN: We predicted “not spam” and it was actually “not spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP**	**FP**
Not Spam (0)	**FN**	**TN**

This kind of contingency table is called **"Confusion Matrix"**

A confusion matrix is better in evaluation in many ways as compared to simple accuracy.

How?

Let's first fill in the confusion matrix with our previous example

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP**	**FP**
Not Spam (0)	**FN**	**TN**

Actual Label	Pred. Label
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

TP: How many predicted “spam” are actually “spam”

A Better Evaluation: Binary Classification

	Actual Labels/Ground Truth	
Predicted Labels/Predictions	Spam (1)	Not Spam (0)
Spam (1)	**TP = 0**	**FP**
Not Spam (0)	**FN**	**TN**

Actual Label	Pred. Label
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

TP: How many predicted “spam” are actually “spam”

FP: How many predicted “spam” are actually “not spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP = 0**	**FP = 1**
Not Spam (0)	**FN**	**TN**

Actual Label	Pred. Label
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

TP: How many predicted “spam” are actually “spam”

FP: How many predicted “spam” are actually “not spam”

FN: How many predicted “not spam” are actually “spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP = 0**	**FP = 1**
Not Spam (0)	**FN = 1**	**TN**

Actual Label	Pred. Label
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

TP: How many predicted “spam” are actually “spam”

FP: How many predicted “spam” are actually “not spam”

FN: How many predicted “not spam” are actually “spam”

TN: How many predicted “not spam” are actually “not spam”

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP = 0**	**FP = 1**
Not Spam (0)	**FN = 1**	**TN = 8**

Actual Label	Pred. Label
0	1
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

TP: How many predicted "spam" are actually "spam"

FP: How many predicted "spam" are actually "not spam"

FN: How many predicted "not spam" are actually "spam"

FN: How many predicted "not spam" are actually "not spam"

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions		Spam (1)	Not Spam (0)
	Spam (1)	**TP = 0**	**FP = 1**
	Not Spam (0)	**FN = 1**	**TN = 8**

Where is accuracy in this confusion matrix?

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = \frac{8}{10} = 0.8$$

FN and FP are errors!

Terminology Alert: FP is called Type-I Error while FN is called Type-II Error

Not true reflexive of prediction performance when class distribution is skewed

Solution: Calculate **Precision** and **Recall** also…

A Better Evaluation: Binary Classification

		Actual Labels/Ground Truth	
		Spam (1)	Not Spam (0)
Predicted Labels/Predictions	Spam (1)	**TP = 0**	**FP = 1**
	Not Spam (0)	**FN = 1**	**TN = 8**

Precision: % of selected items that are correct (or in simple terms, percentage of true positives amongst all predicted positives)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{0}{0 + 1} = 0$$

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions		Spam (1)	Not Spam (0)
	Spam (1)	**TP = 0**	**FP = 1**
	Not Spam (0)	**FN = 1**	**TN = 8**

Recall: % of correct items that are selected (or in simple terms, percentage of positives that you were able to find/predict)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{0}{0 + 1} = 0$$

A Better Evaluation: Binary Classification

An extreme example of imbalanced dataset: Out of 1000, only 10 are "spams".

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP = 0**	**FP = 0**
Not Spam (0)	**FN = 10**	**TN = 990**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = \frac{0 + 990}{1000} = 0.99$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{0}{0 + 0} = 0$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{0}{0 + 10} = 0$$

Let's revise our model...

A Better Evaluation: Binary Classification

An extreme example of imbalanced dataset: Out of 1000, only 10 are “spams”.

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP = 8**	**FP = 30**
Not Spam (0)	**FN = 2**	**TN = 960**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = \frac{8 + 960}{1000} = 0.968$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{8}{8 + 30} = 0.21$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{8}{8 + 2} = 0.8$$

Recall improved but at a cost. The model predicts “spam” for many emails that are “not spam”!

What’s more important to you? Recall or Precision?

We can combine Precision and Recall to assess tradeoff between the two.

A Better Evaluation: Binary Classification

Actual Labels/Ground Truth

Predicted Labels/Predictions

	Spam (1)	Not Spam (0)
Spam (1)	**TP = 8**	**FP = 30**
Not Spam (0)	**FN = 2**	**TN = 960**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} = \frac{8 + 960}{1000} = 0.968$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{8}{8 + 30} = 0.21$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{8}{8 + 2} = 0.8$$

$$F1 = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} = \frac{0.42}{1.01} = 0.33$$

What about multiclass problem?

Summary

	Gold Positive	Gold Negative	
Predicted Positive	**True Positives (TP)**	**False Positives (FP)**	$\frac{TP}{TP+FP}$ "Precision" or "Positive Predictive Value"
Predicted Negative	**False Negatives (FN)**	**True Negatives (TN)**	$\frac{TN}{FN+TN}$ "Negative Predictive Value"
	$\frac{TP}{TP+FN}$ "Recall" or "Sensitivity" or "True Positive Rate"	$\frac{TN}{FP+TN}$ "Specificity" or "True Negative Rate"	

Suppose you go for Covid test. The test report says you are Covid positive. But it also mentions Sensitivity value of the machine which is very low. How reliable the result is?

Suppose you go for Covid test. The test report says you are Covid negative. But it also mentions Specificity value of the machine which is very low. How reliable the result is?

Summary

Accuracy

- What fraction of time am I correct in my classification?

$$\frac{My\ Correct\ Answers}{All\ Questions} = \frac{TP}{TP + TN + FP + FN}$$

Recall (Sensitivity)

- How much of the reality has been covert by my positive output?
- What fraction of true positives is captured by my positives?

$$\frac{True\ Positives}{Real\ Positives} = \frac{TP}{TP + FN}$$

Precision

- How much should you trust me when I say something tests positive
- What fraction of my positives are true positives

$$\frac{True\ Positives}{My\ Positives} = \frac{TP}{TP + FP}$$

Specificity

- How much of the reality has been covered by my negative output?
- What fraction of the true negatives is captured by my negatives?

$$\frac{True\ Negatives}{Real\ Negatives} = \frac{TN}{TN + FP}$$

Precision Recall Tradeoff

- **Cancer Detection**
  - Recall is more important (Detect all cancer patients, even if there are false positives)
- **Information Retrieval**
  - Precision is more important (Whatever is retrieved should be relevant, even if a few relevant records are missed (false negatives))
- **Death Sentence through ML**
  - Precision is more important (It’s okay to miss a punishment than incriminating an innocent (false positives))
- **Spam Email Detection**
  - Precision is more important (It’s okay to miss out a spam email, but no ham emails should be filtered out (false positives))
- **Detecting Fraudulent Bank Transactions**
  - Recall is more important (It’s okay to classify a legitimate transaction as fraudulent than to classify a fraudulent transaction as legit (false negatives))

Precision Recall Tradeoff

- **High recall but low precision**
  - Most ground-truth labels are correctly predicted but most predictions are incorrect (many false positives).
- **High precision but low recall**
  - Most predictions are correct, but most ground-truth labels are not detected (many false negatives).
- **High precision and high recall**
  - Ideal case where all ground-truth labels are predicted correctly and there was no false positive
- **Low precision and low recall**
  - Least desirable predictor that does not detect most ground-truth objects (many false negatives), and most detections are incorrect (many false positives).

Other Performance Evaluation Metrics

❑ F-Measure

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- The β parameter weights the importance of recall and precision
  - Based on the needs of an application
  - Values of $\beta > 1$ favor recall, while
  - Values of $\beta < 1$ favor precision.

❑ When $\beta = 1$, precision and recall are equally balanced.

- This is the most frequent used metric and as we saw it earlier, is called $F_\beta = 1$ score or just F1:

$$F_1 = \frac{(1+1)PR}{1P + R} = \frac{2 \times P \times R}{P + R}$$

Using **GINI Index** for Decision Tree Construction

Measure of Impurity: GINI

- GINI Index for a given node t:

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

Where $p(j|t)$ is the relative frequency of class j at node t

- ***Maximum*** $\left(1 - \frac{1}{n_c}\right)$ when records are equally distributed among all classes, implying ***least information***
- ***Minimum (0)*** when all records belong to one class, implying ***most 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	

Note: Behavior of GINI is just like Entropy with respect to class label distribution (i.e., minimum if only one class exists at a node, and maximum if both classes are equal in numbers)

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, SLIQ, SPRINT
- When a 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 is the number of records at child i

n is the number of records at node p

Let's quickly see how GINI can be helpful in splitting/merging/thresholding of Binary, Categorical, and Numerical Attributes.

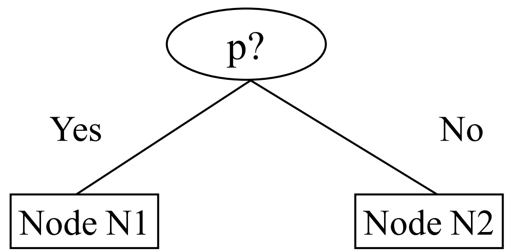
Splitting Based on GINI: Binary Attributes

- Split the node p into two partitions
  - ***Larger*** and ***Purer Partitions*** are sought for.

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

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

	Parent
C1	**6**
C2	**6**
Gini = 0.500	



	N1	**N2**
C1	**5**	**1**
C2	**2**	**4**
Gini=	**0.333**	

$$GINI(N1) = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 0.194$$

$$GINI(N2) = 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{4}{6}\right)^2 = 0.528$$

$$GINI(Split) = \frac{7}{12} \times 0.194 + \frac{5}{12} \times 0.528 = 0.333$$

Splitting Based on GINI: Categorical Attributes

- For each distinct value, gather counts for each class in the dataset
- Use this count matrix to make decisions
- Should you merge two values in an attribute?

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

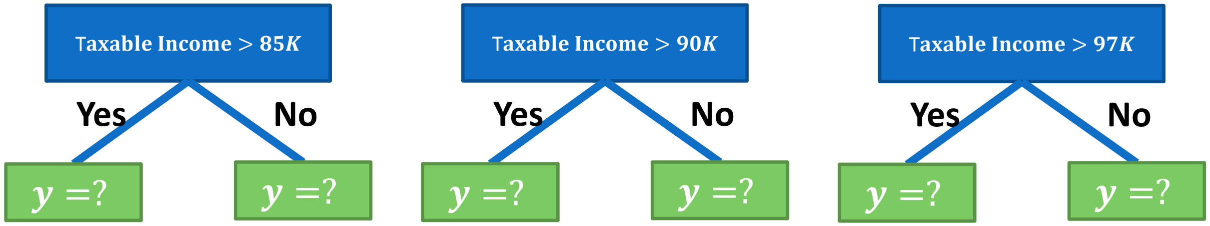
	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Which split is better?

Splitting Based on GINI: Continuous Attributes

- Use Binary Decisions based on one value (after thresholding)
- **What should be splitting threshold value?**
- Several choices for the splitting value
  - Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
  - Class counts in each of the partitions $A < v$ and $A \geq v$
- Simple method to choose best v
  - For each v, scan the database to gather count matrix and compute its GINI Index
  - Computationally inefficient, repetition of work

Tid	Refund	Marital Status	Taxable Income	y
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



Splitting Based on GINI: Continuous Attributes

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing GINI Index
  - Choose the split position that has the least GINI Index

Sorted Values

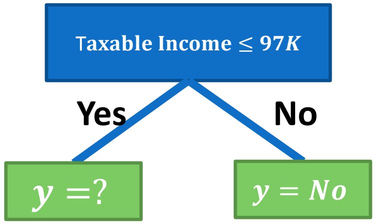
y	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income	60	70	75	85	90	95	100	120	125	220

Split Positions

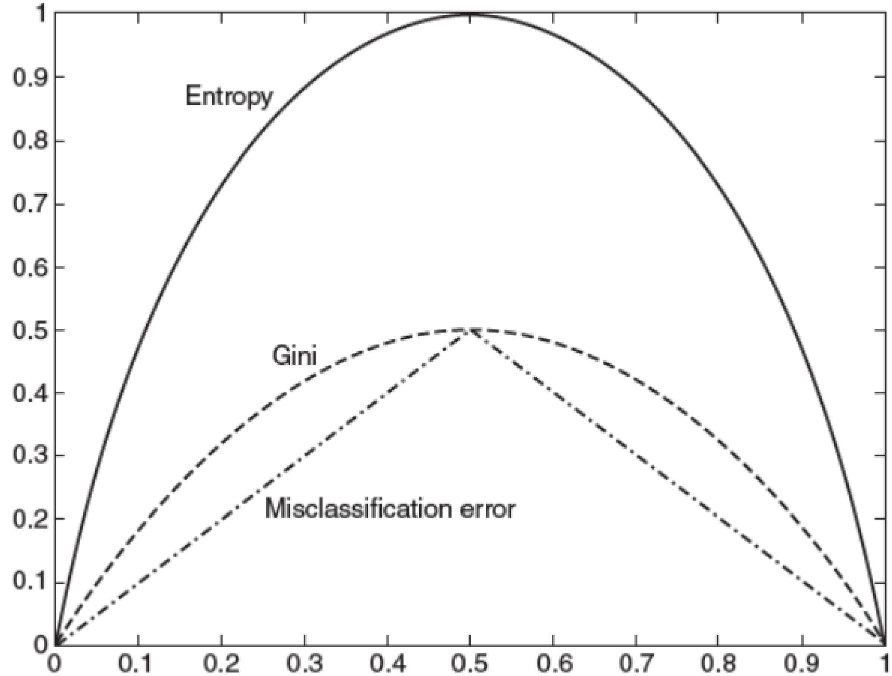
	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

Which split is better?

The one with minimum GINI.



GINI vs Entropy



Book Reading

- Murphy – Chapter 1, Chapter 14
- Tom Mitchel (TM) – Chapter 3