**6.3**

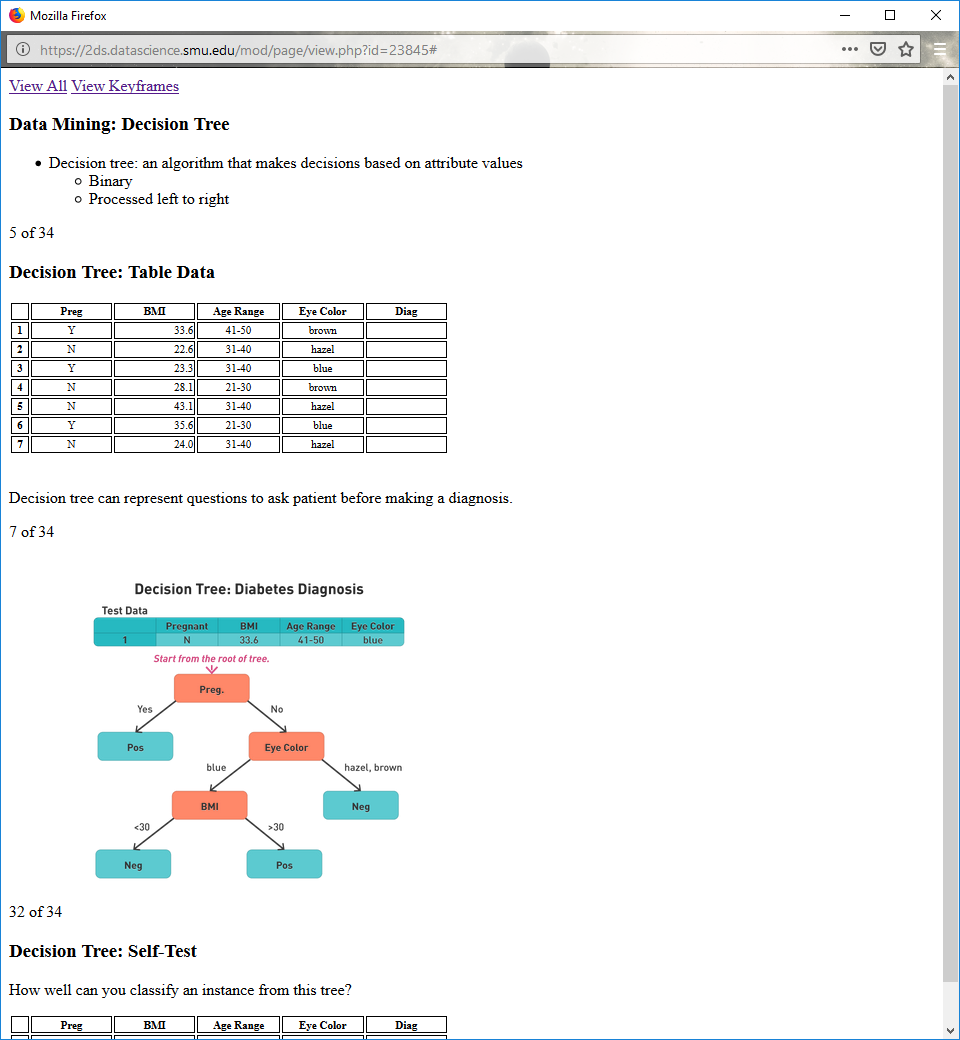
**Data Mining: Decision Tree**

Decision tree: an algorithm that makes decisions based on attribute values

Binary

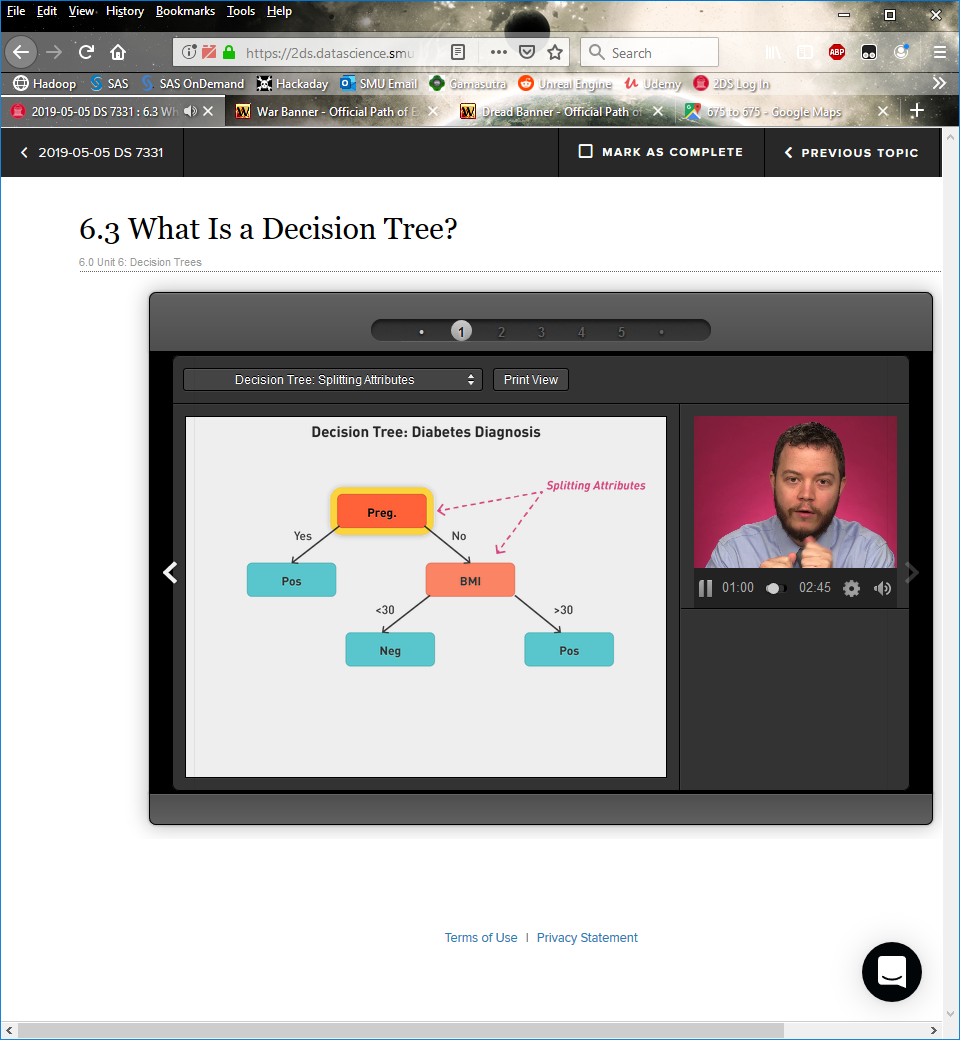
Processed left to right

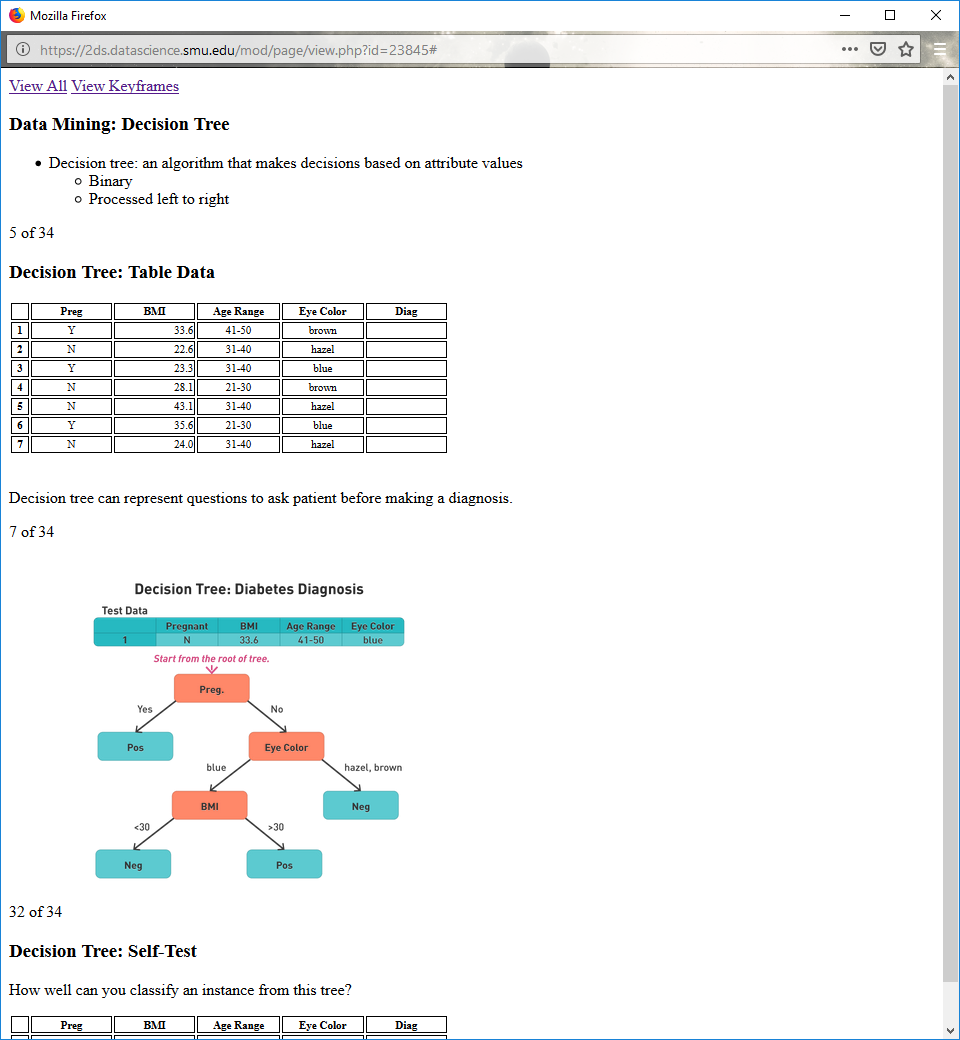
**Decision Tree: Table Data**



Decision tree can represent questions to ask patient before making a diagnosis.

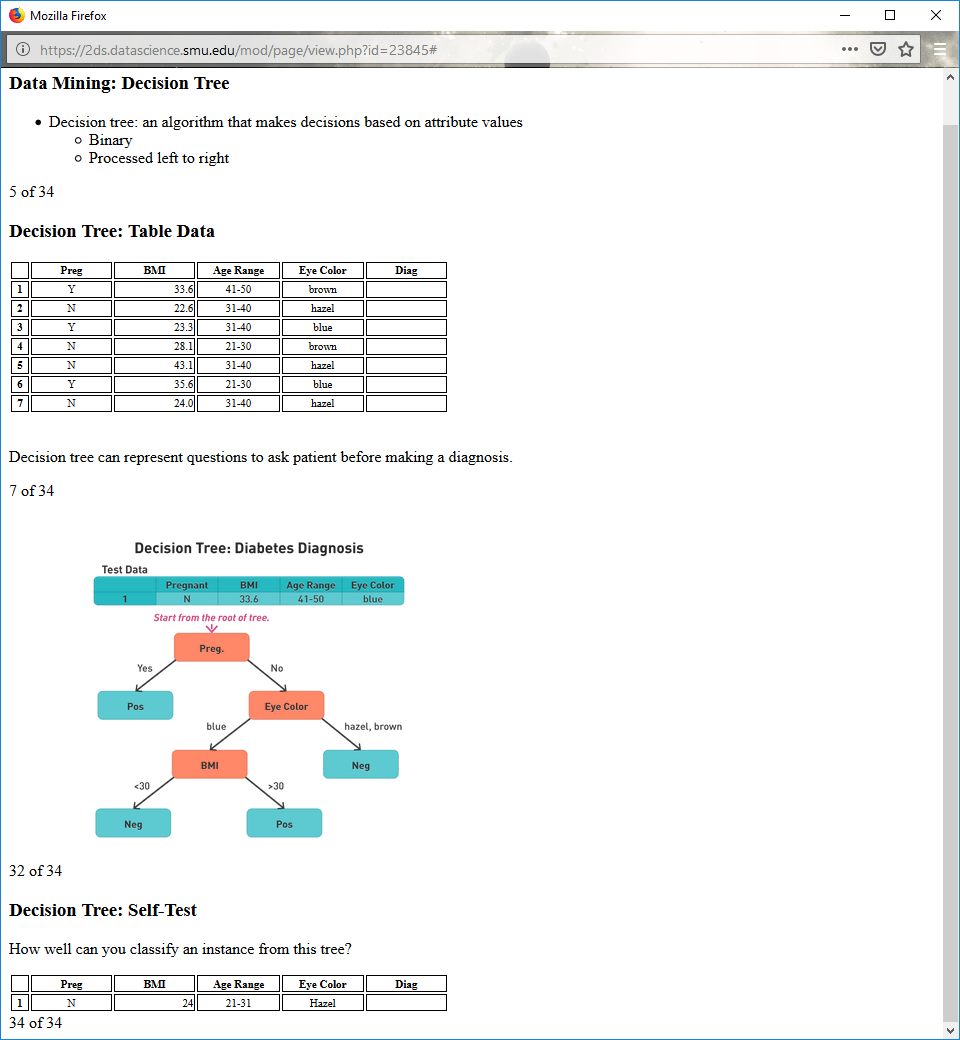
**Decision Tree: Diabetes Diagnosis**



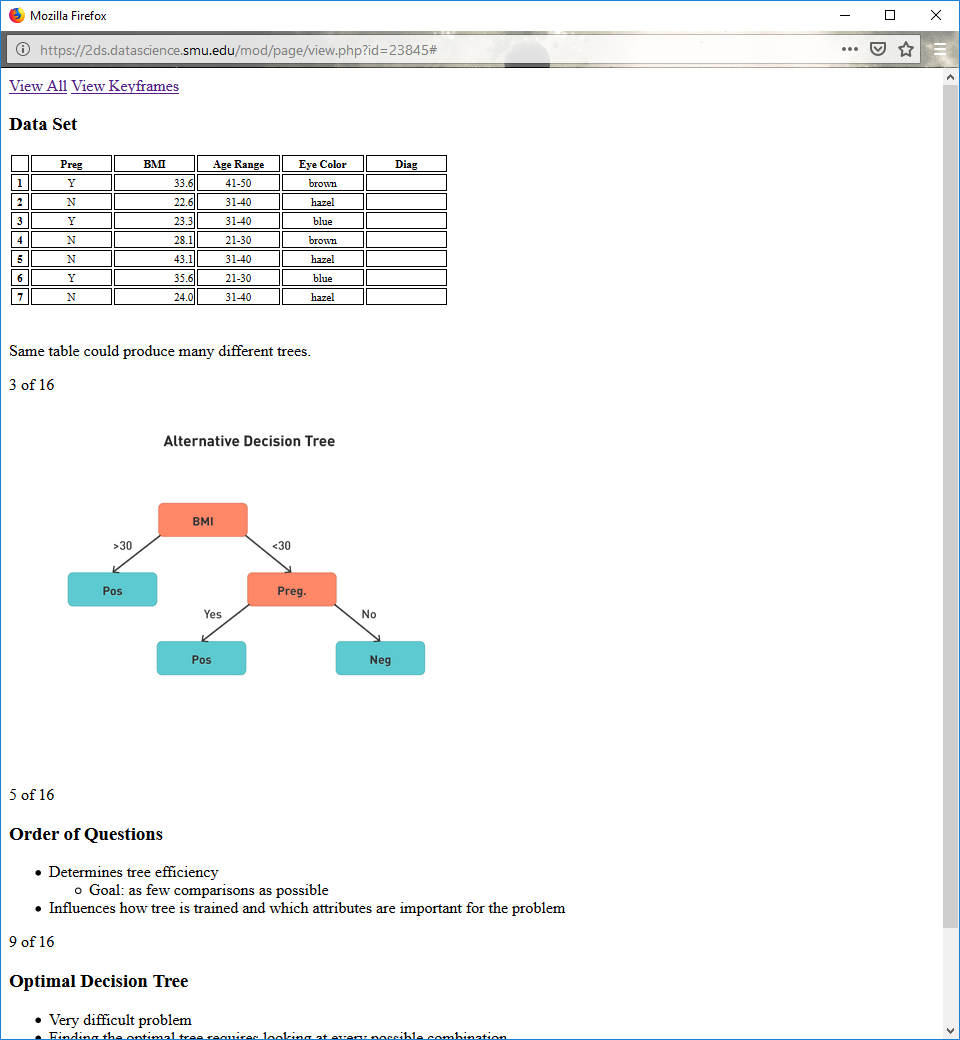


**Decision Tree: Self-Test**

How well can you classify an instance from this tree?

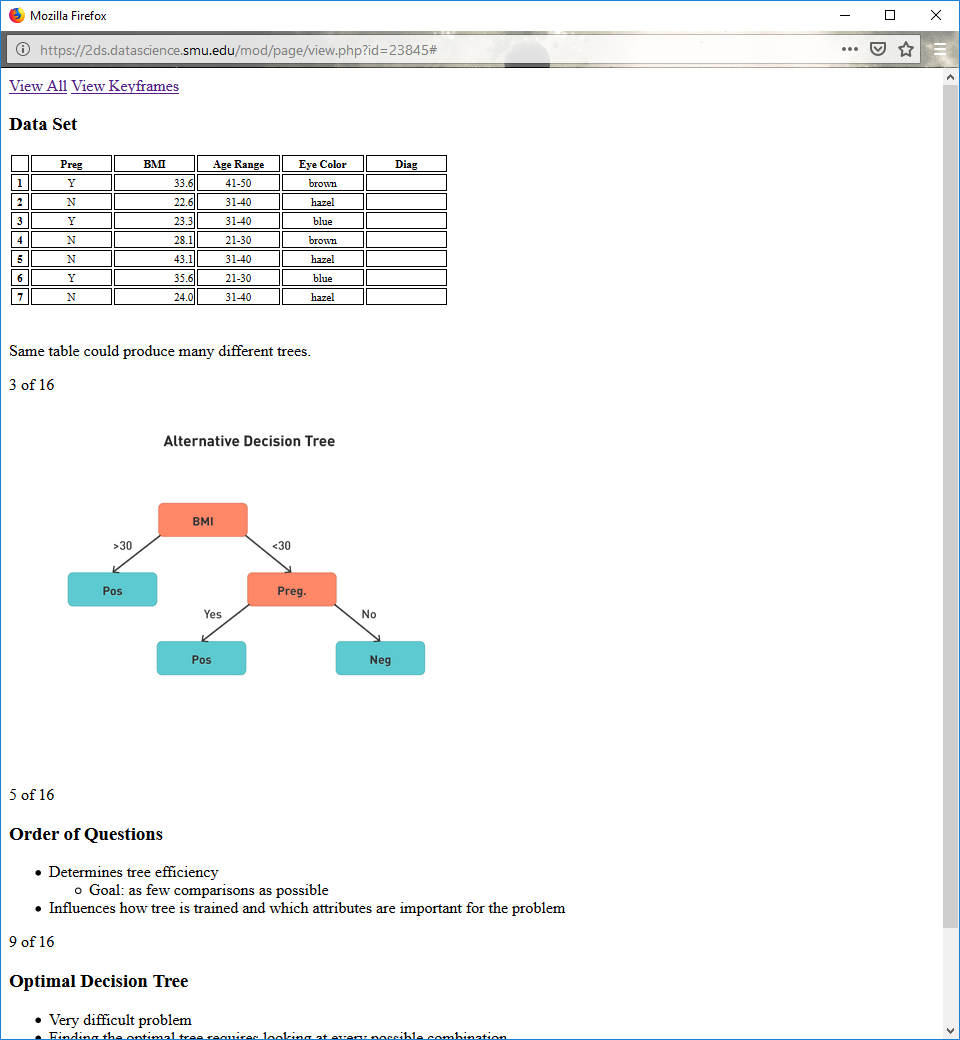


**Data Set**



Same table could produce many different trees

**Alternative Decision Tree**



**Order of Questions**

Determines tree efficiency

Goal: as few comparisons as possible

Influences how tree is trained and which attributes are important for the problem

**Optimal Decision Tree**

Very difficult problem

Finding the optimal tree requires looking at every possible combination

NP-hard problem

Must come up with different methods to split data

Suboptimal but will still work

**6.4**

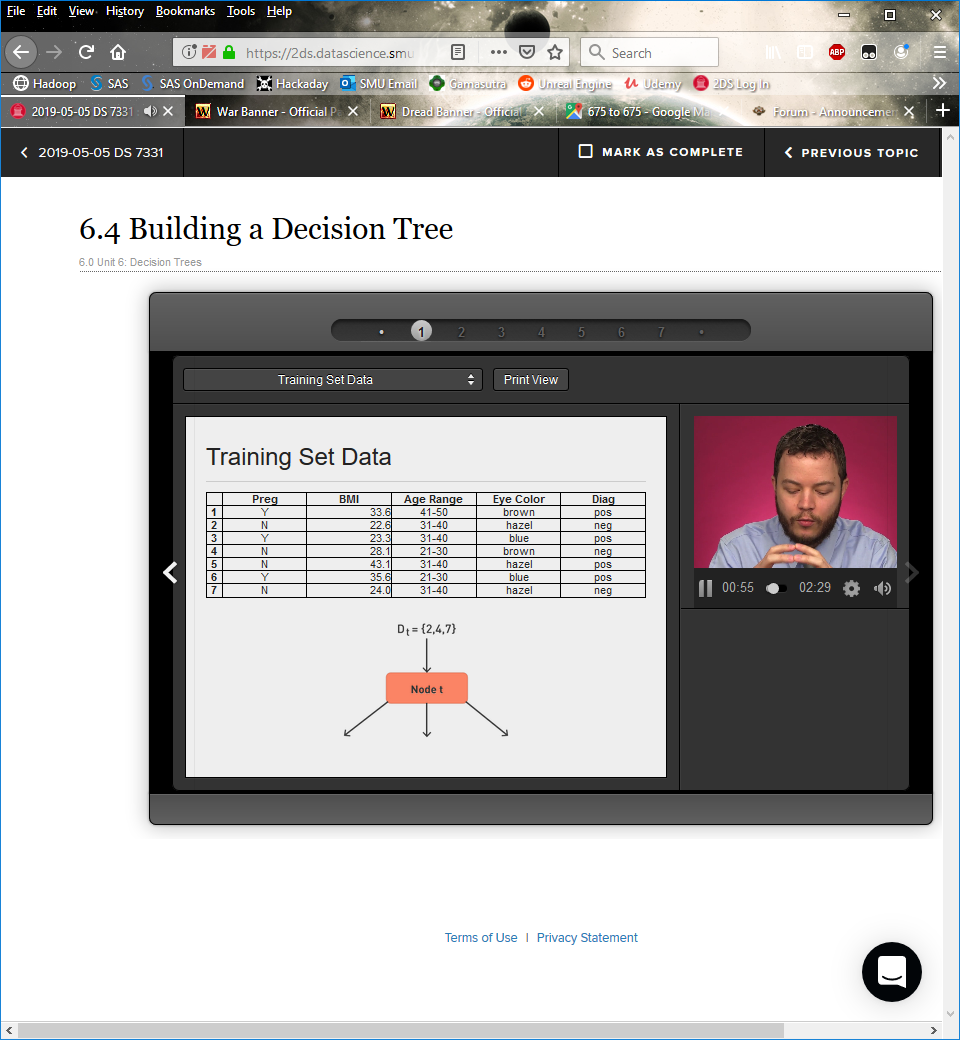
**Decision Tree Induction**

Many common algorithms

Iterative dichotomiser (ID3), C4.5, J48

All build from Hunt's algorithm

**Training Set Data**



**Hunt's Algorithm**

Let Dt be the set of training records that reach a node t

If all records in Dt are the same class yt

Then t is a leaf node labeled as yt

If Dt is an empty set

Then t is a leaf node labeled by the default class, yd

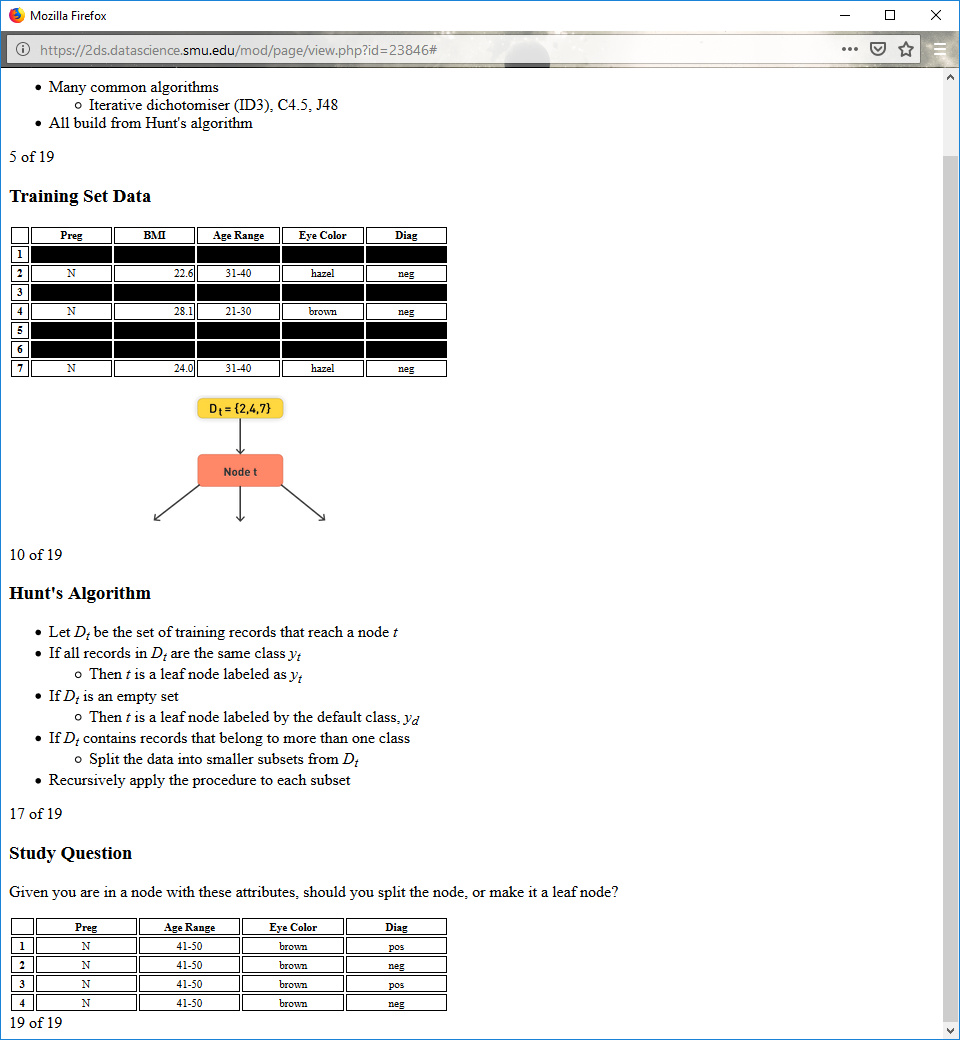
If Dt contains records that belong to more than one class

Split the data into smaller subsets from Dt

Recursively apply the procedure to each subset

**Study Question**

Given you are in a node with these attributes, should you split the node, or make it a leaf node?



**Tree Induction**

How will the data be split?

How will the test condition be specified?

**Splitting the Data**

Depends on attribute types

Nominal

Ordinal

Continuous

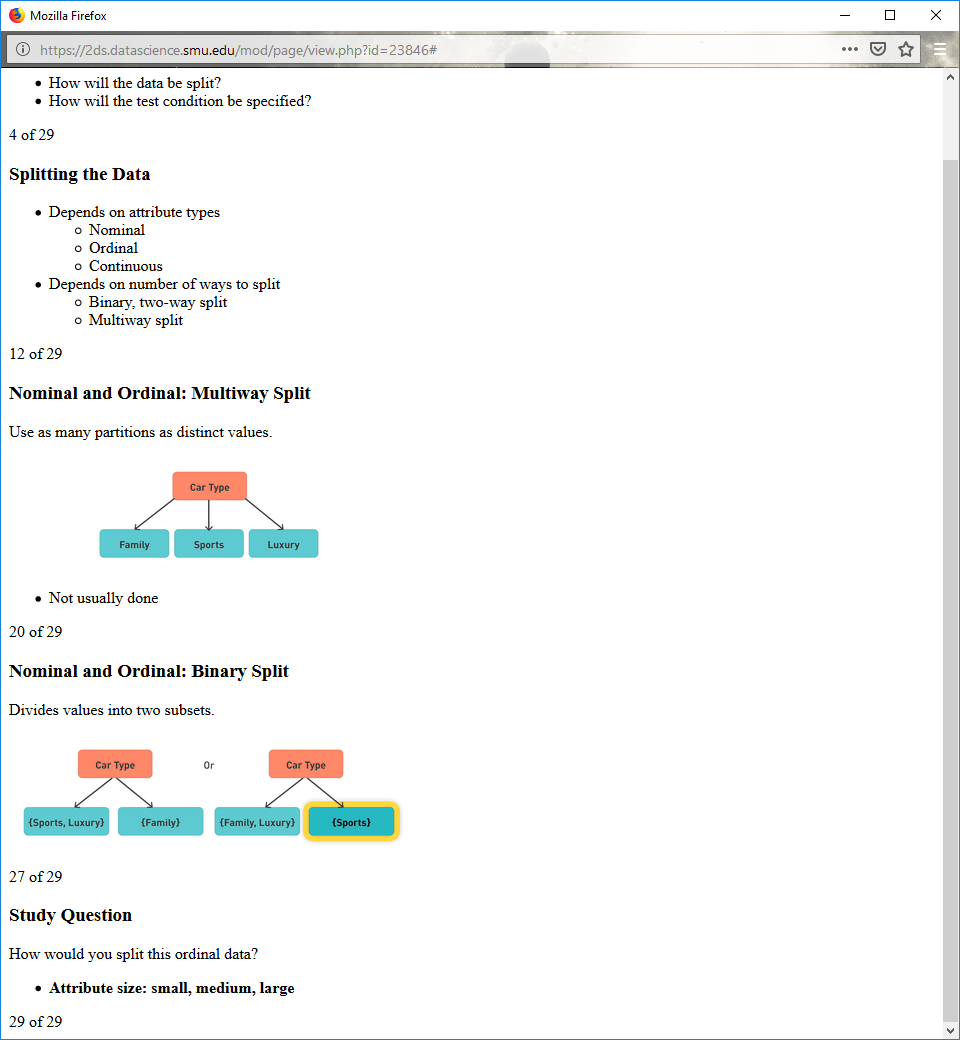
Depends on number of ways to split

Binary, two-way split

Multiway split

**Nominal and Ordinal: Multiway Split**

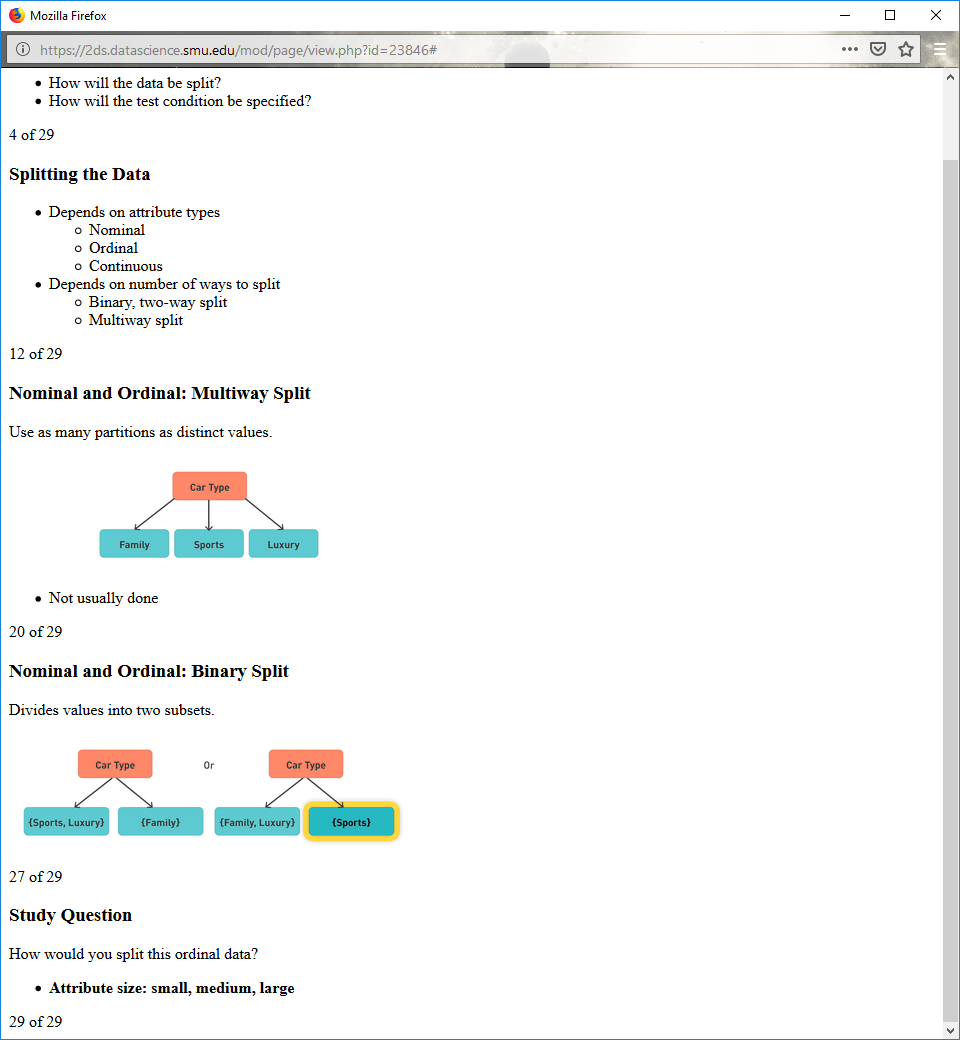
Use as many partitions as distinct values.



Not usually done

**Nominal and Ordinal: Binary Split**

Divides values into two subsets.



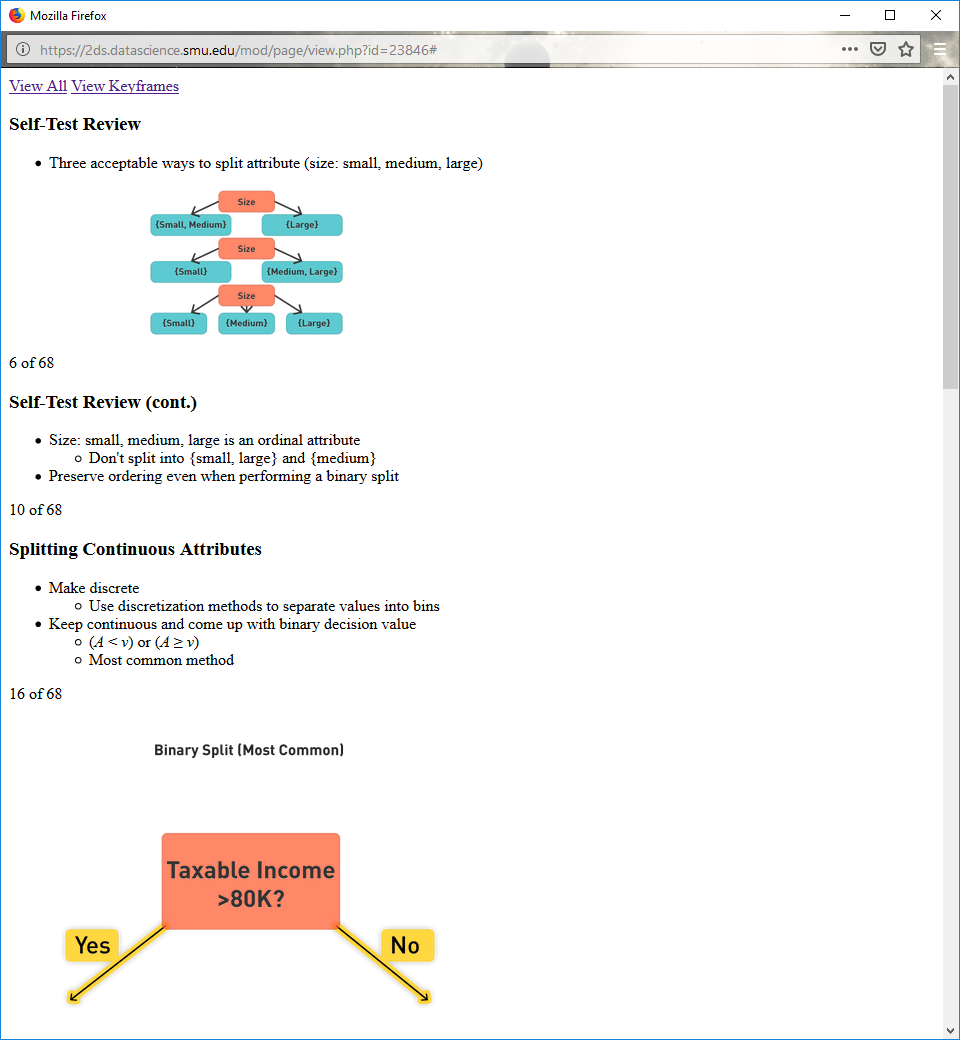
**Study Question**

How would you split this ordinal data?

**Attribute size: small, medium, large**

**Self-Test Review**

Three acceptable ways to split attribute (size: small, medium, large)



**Self-Test Review (cont.)**

Size: small, medium, large is an ordinal attribute

Don't split into {small, large} and {medium}

Preserve ordering even when performing a binary split

**Splitting Continuous Attributes**

Make discrete

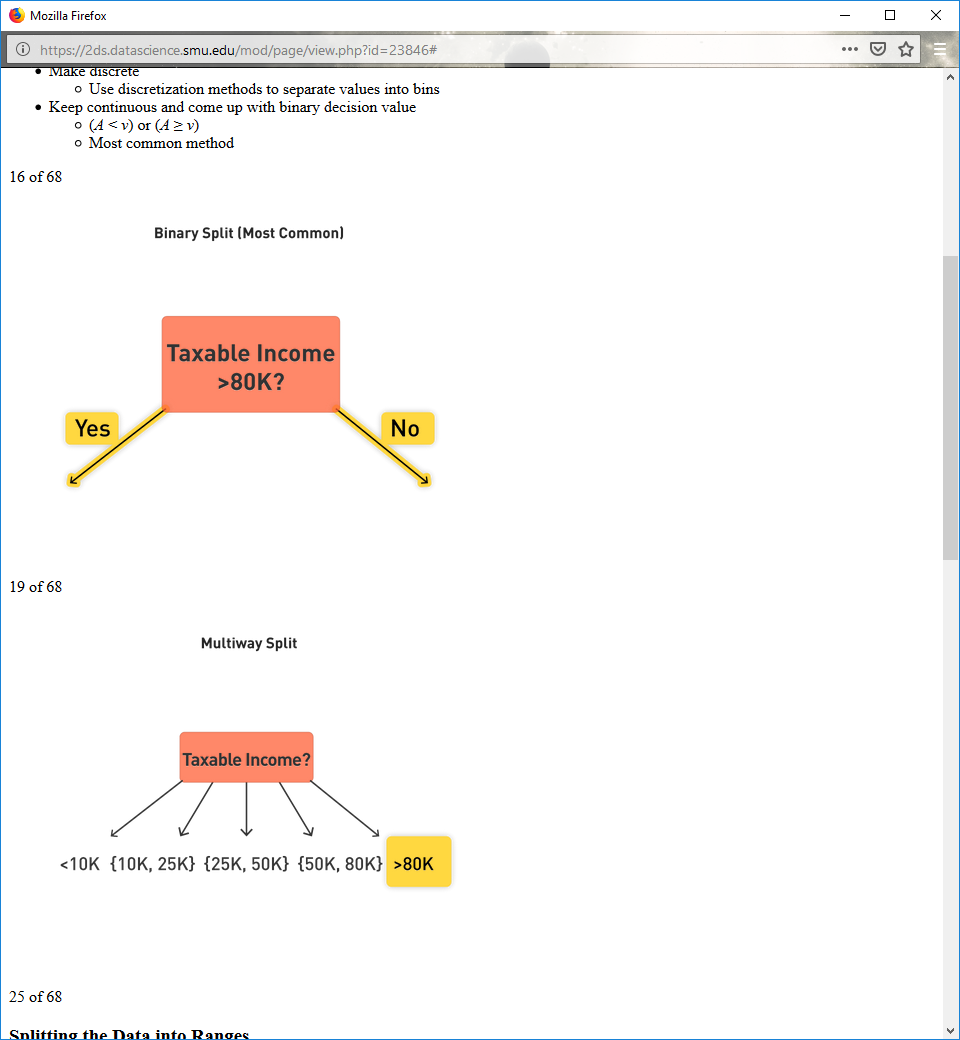
Use discretization methods to separate values into bins

Keep continuous and come up with binary decision value

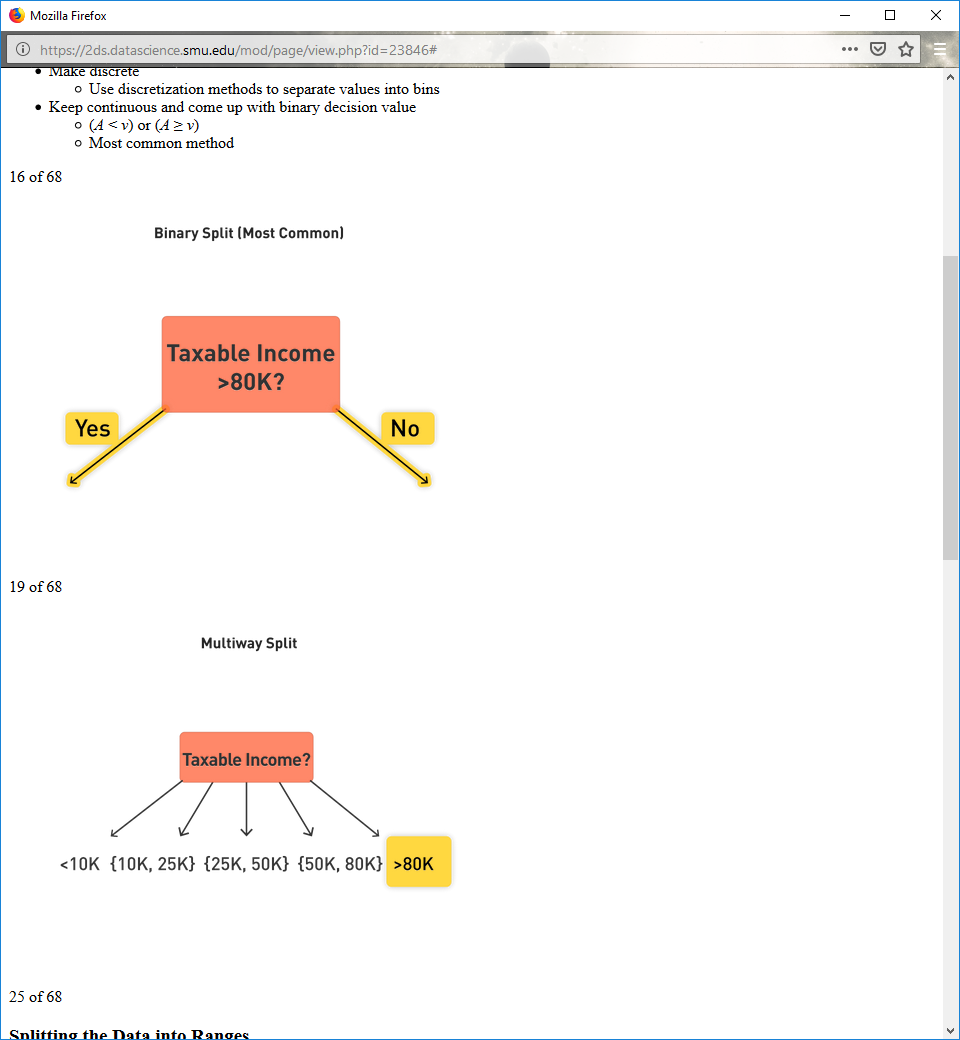
(A < v) or (A ≥ v)

Most common method

**Binary Split (Most Common)**



**Multiway Split**



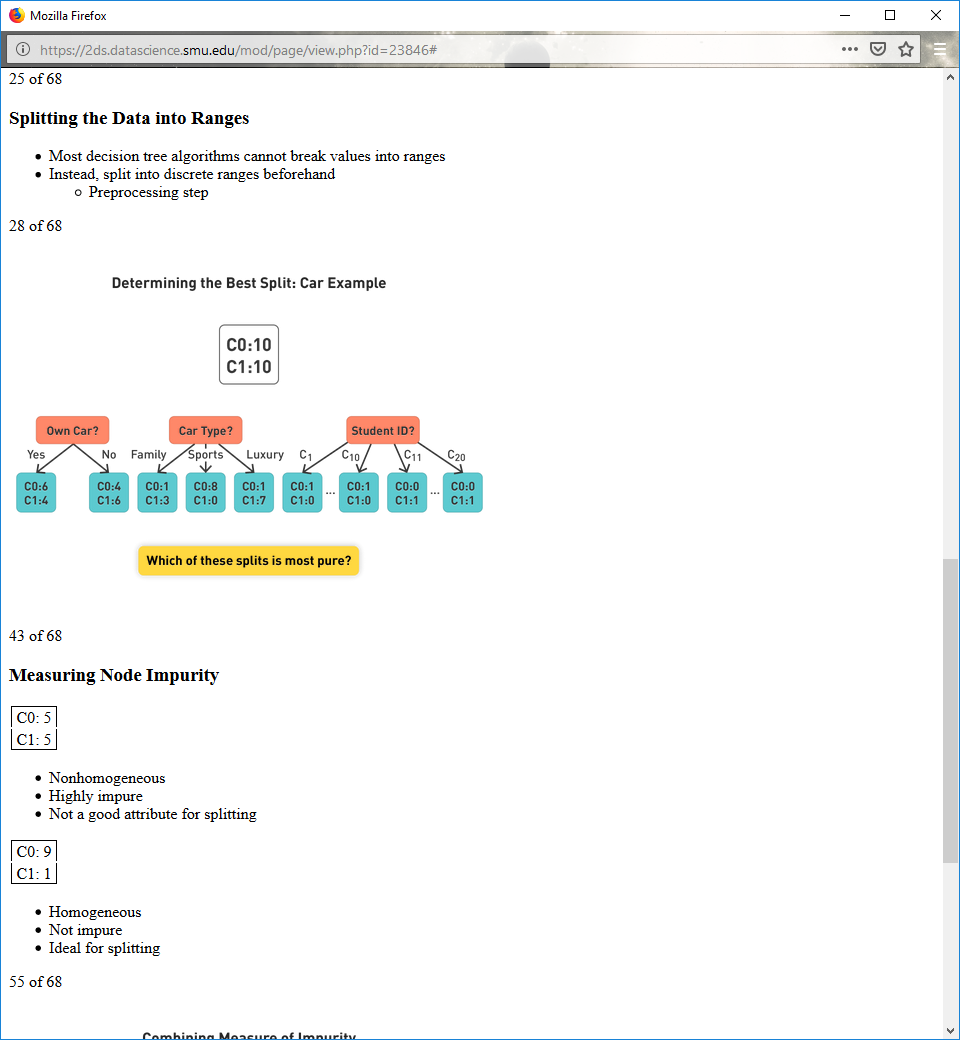
**Splitting the Data into Ranges**

Most decision tree algorithms cannot break values into ranges

Instead, split into discrete ranges beforehand

Preprocessing step

**Determining the Best Split: Car Example**



**Measuring Node Impurity**

|  |
| --- |
| C0: 5 |
| C1: 5 |

Nonhomogeneous

Highly impure

Not a good attribute for splitting

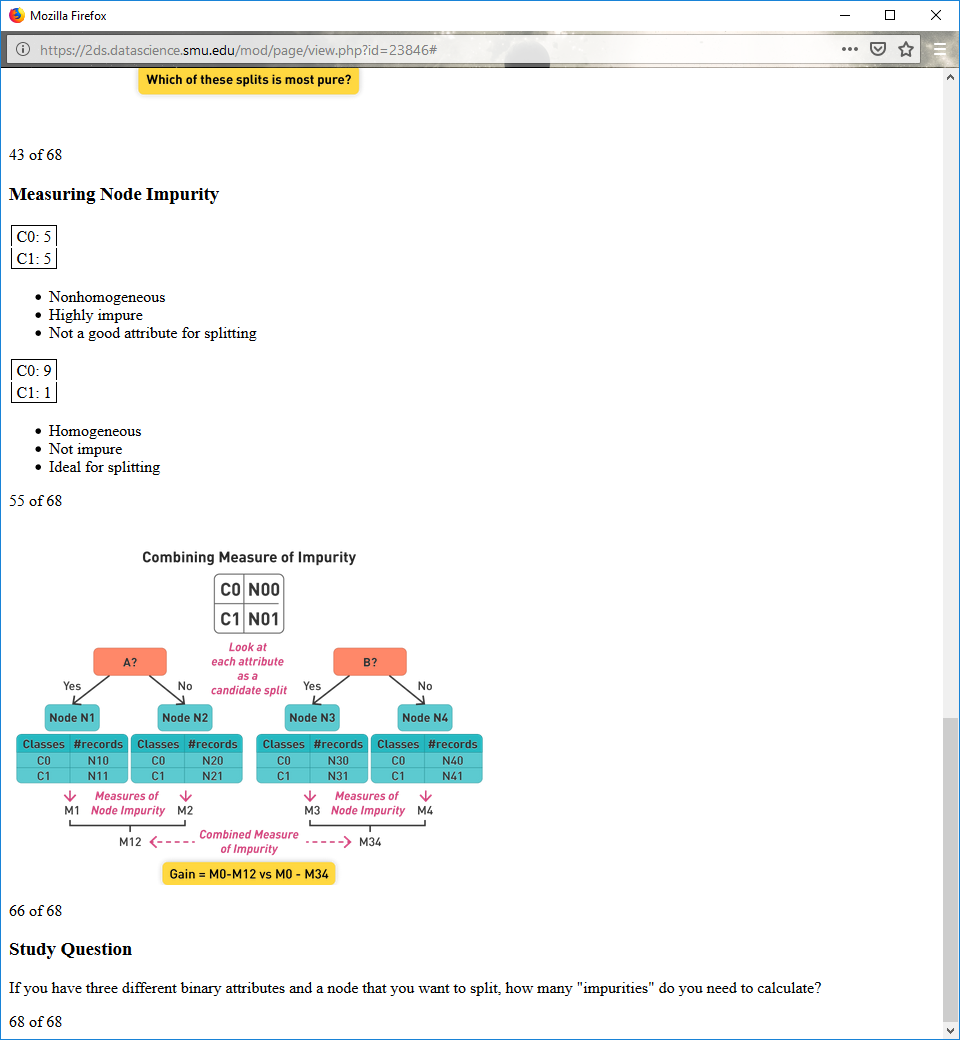
|  |
| --- |
| C0: 9 |
| C1: 1 |

Homogeneous

Not impure

Ideal for splitting

**Combining Measure of Impurity**



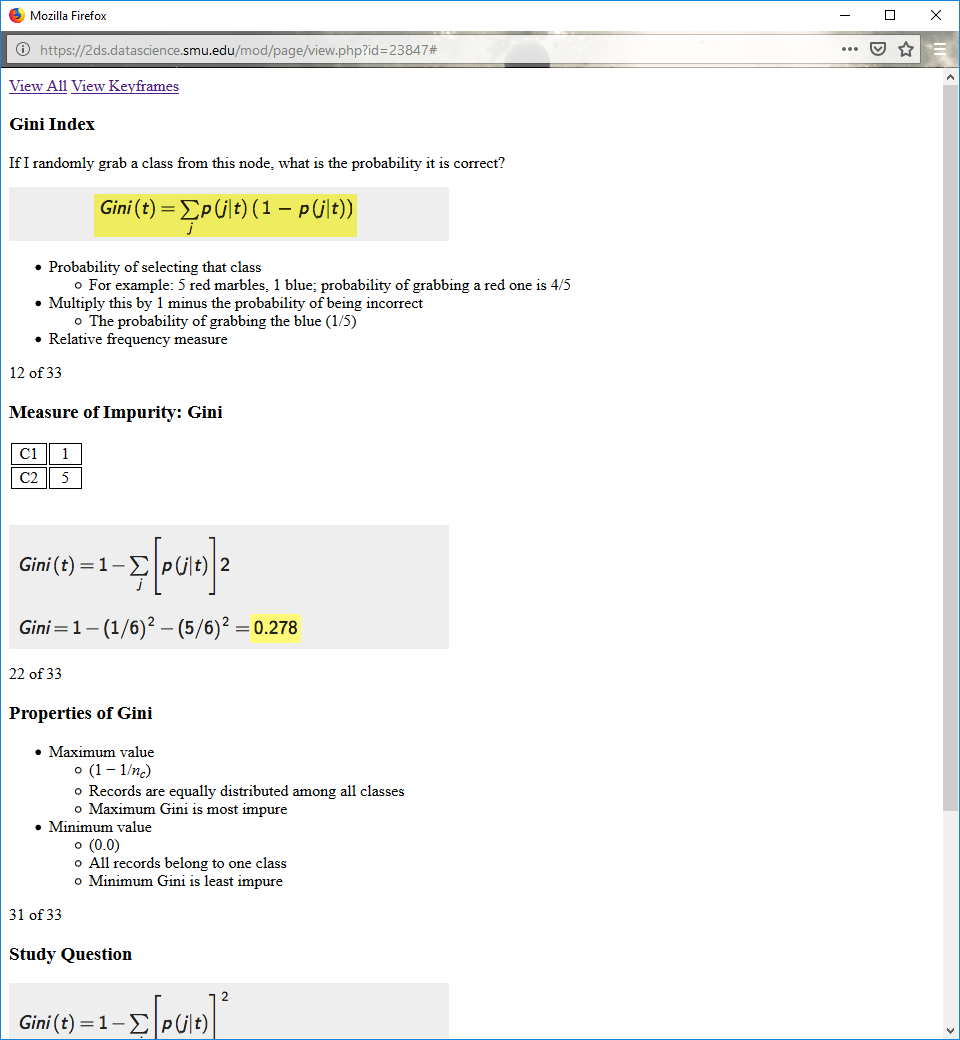
**Study Question**

If you have three different binary attributes and a node that you want to split, how many "impurities" do you need to calculate?

**6.5**

**Gini Index**

If I randomly grab a class from this node, what is the probability it is correct?



Probability of selecting that class

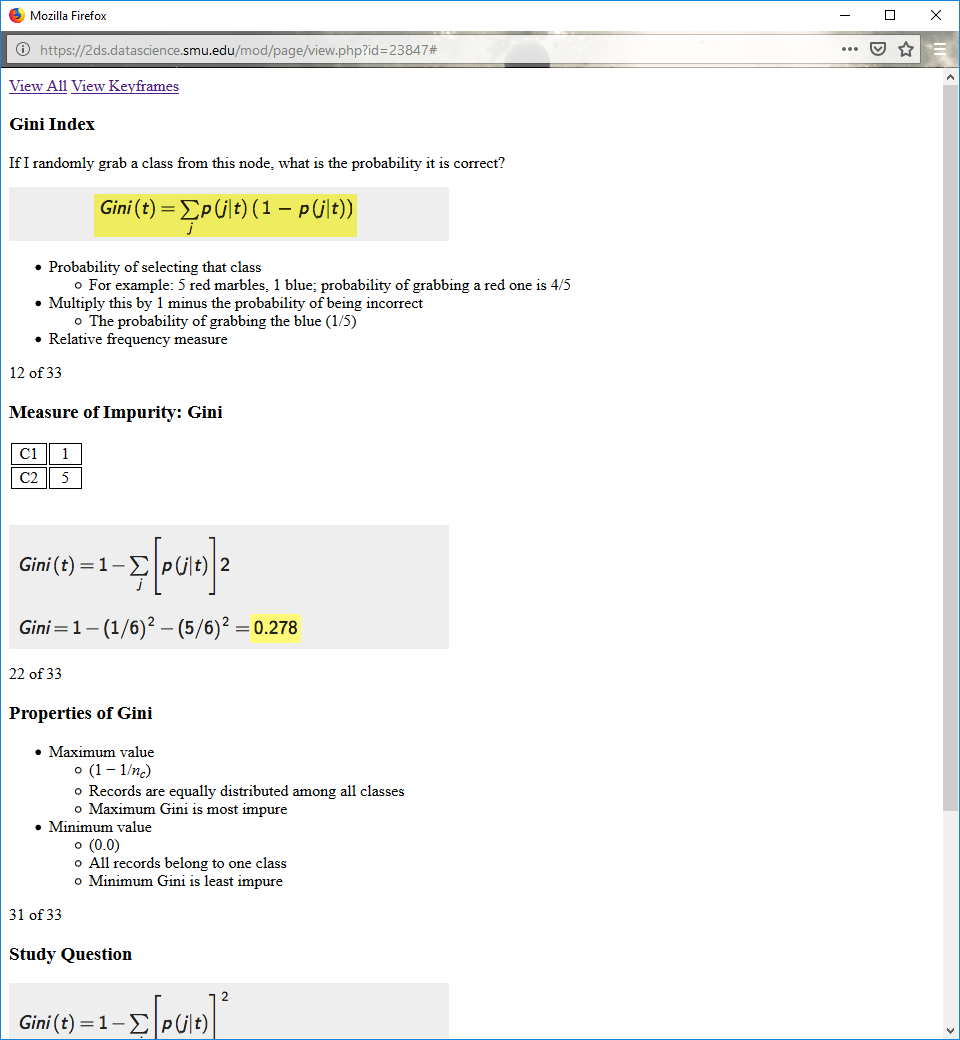
For example: 5 red marbles, 1 blue; probability of grabbing a red one is 4/5

Multiply this by 1 minus the probability of being incorrect

The probability of grabbing the blue (1/5)

Relative frequency measure

**Measure of Impurity: Gini**



**Properties of Gini**

Maximum value

(1 − 1/nc) n is the number of classes

Records are equally distributed among all classes

Maximum Gini is most impure

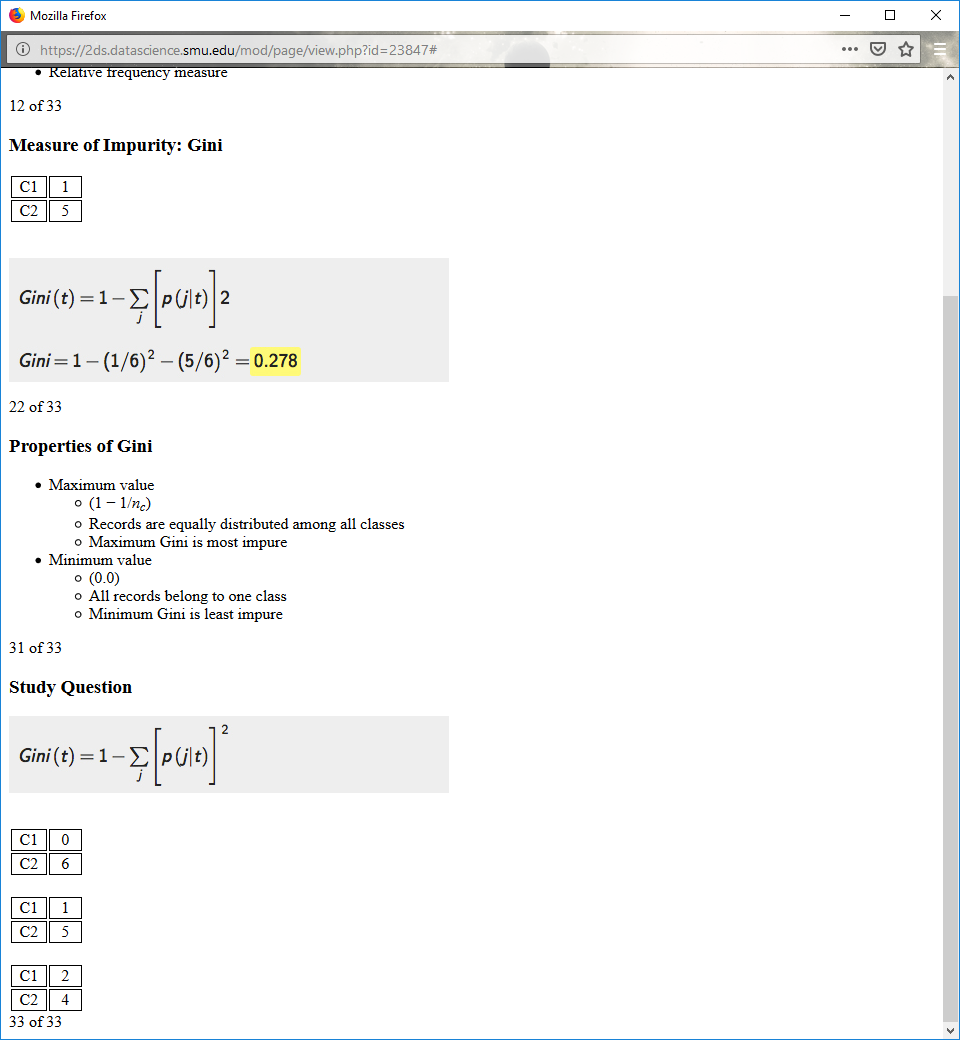
Minimum value

(0.0)

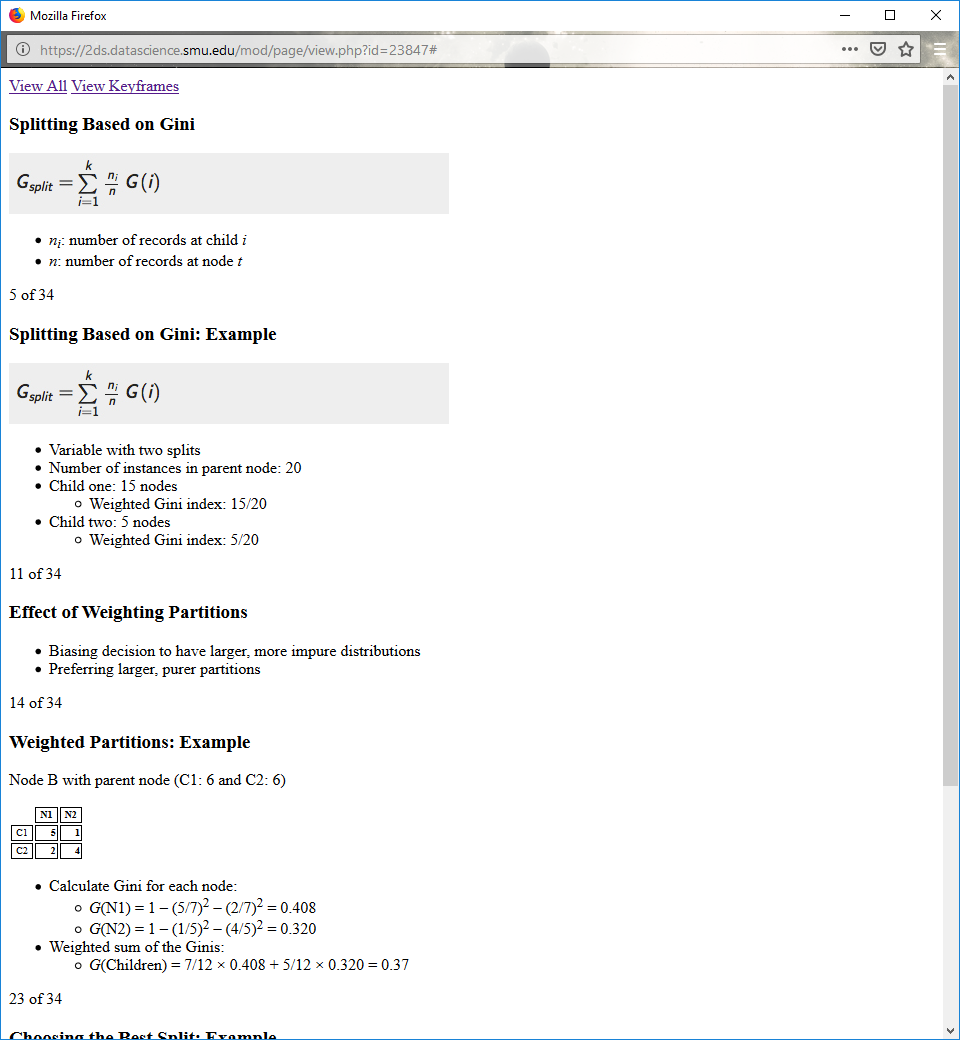
All records belong to one class

Minimum Gini is least impure

**Study Question**



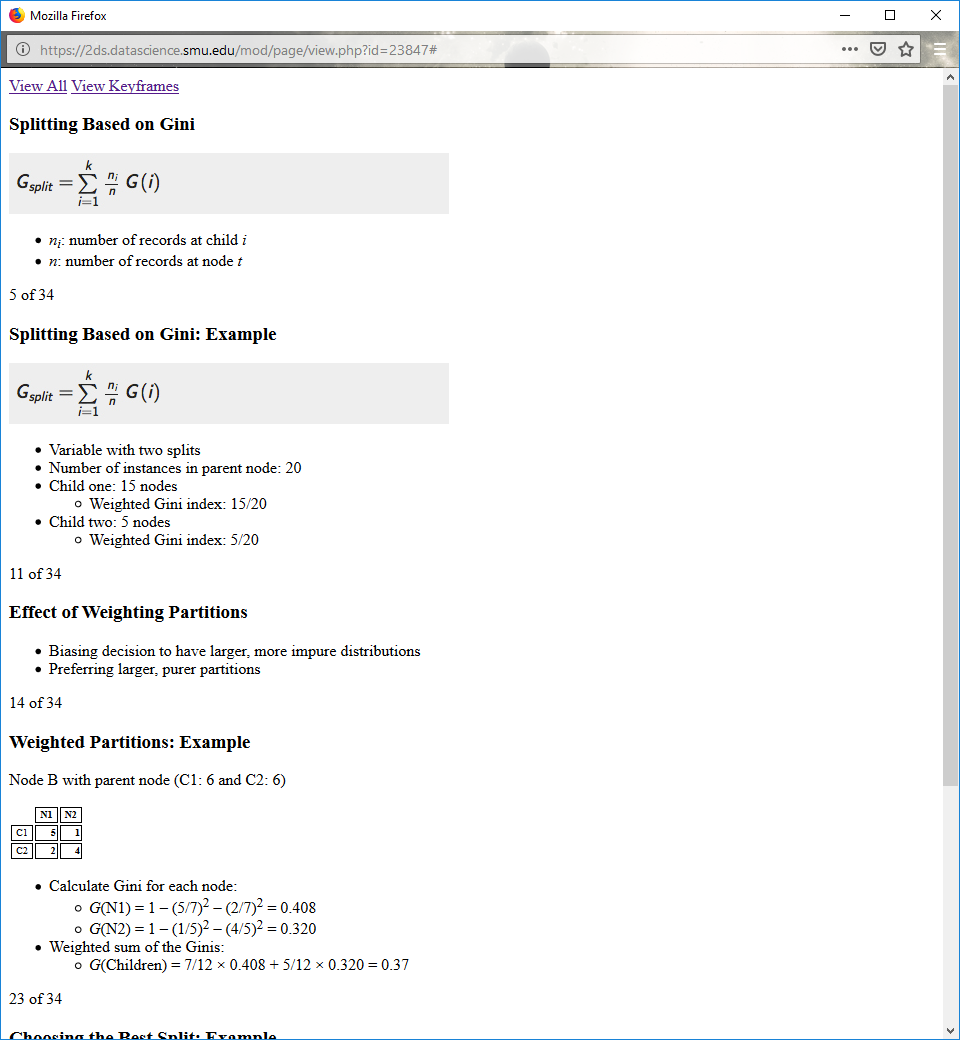
**Splitting Based on Gini**



ni: number of records at child i

n: number of records at node t

**Splitting Based on Gini: Example**



Variable with two splits

Number of instances in parent node: 20

Child one: 15 nodes

Weighted Gini index: 15/20

Child two: 5 nodes

Weighted Gini index: 5/20

**Effect of Weighting Partitions**

Biasing decision to have larger, more impure distributions

Preferring larger, purer partitions

**Weighted Partitions: Example**

Node B with parent node (C1: 6 and C2: 6)

|  |  |  |
| --- | --- | --- |
|  | **N1** | **N2** |
| C1 | **5** | **1** |
| C2 | **2** | **4** |

Calculate Gini for each node:

G(N1) = 1 – (5/7)2 – (2/7)2 = 0.408

G(N2) = 1 – (1/5)2 – (4/5)2 = 0.320

Weighted sum of the Ginis:

G(Children) = 7/12 × 0.408 + 5/12 × 0.320 = 0.37

**Choosing the Best Split: Example**

Calculating with an ordinal attribute:

Two-way splits

{Medium, Small} {Large}

{Small} {Medium, Large}

Calculate weighted Gini for each

Multiway split

{Small} {Medium} {Large}

Calculate and compare weighted Ginis for 3-node and 2-node options

**Study Question**

Does a multiway split always produce a lower Gini?

**Study Question Review**

**Question:** Does a multiway split always produce a lower Gini?  
**Answer:** Yes, when split into more nodes, you will always get a lower Gini index.

**Computing Continuous Attributes**

Divide data based on threshold values

Use binary decisions

Many choices for splitting a continuous valued variable

Trim down number of choices

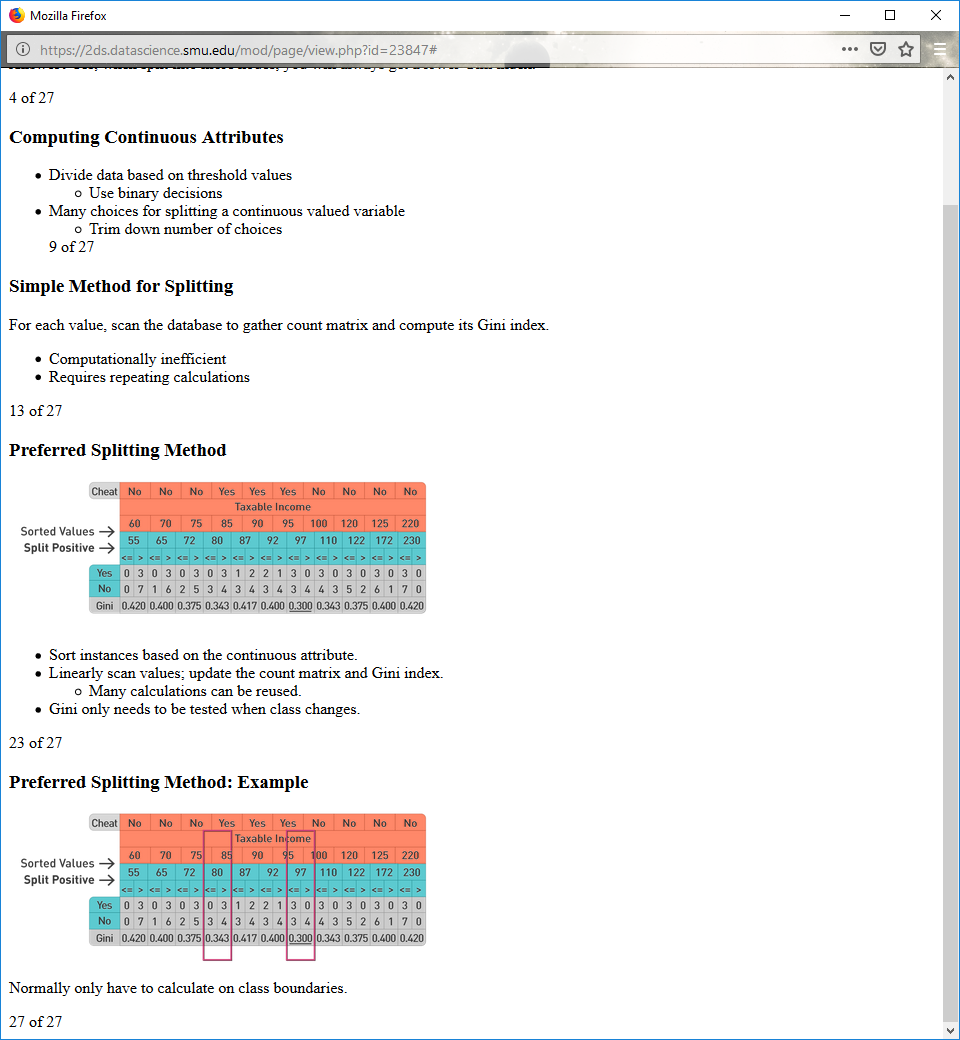
**Simple Method for Splitting**

For each value, scan the database to gather count matrix and compute its Gini index.

Computationally inefficient

Requires repeating calculations

**Preferred Splitting Method**



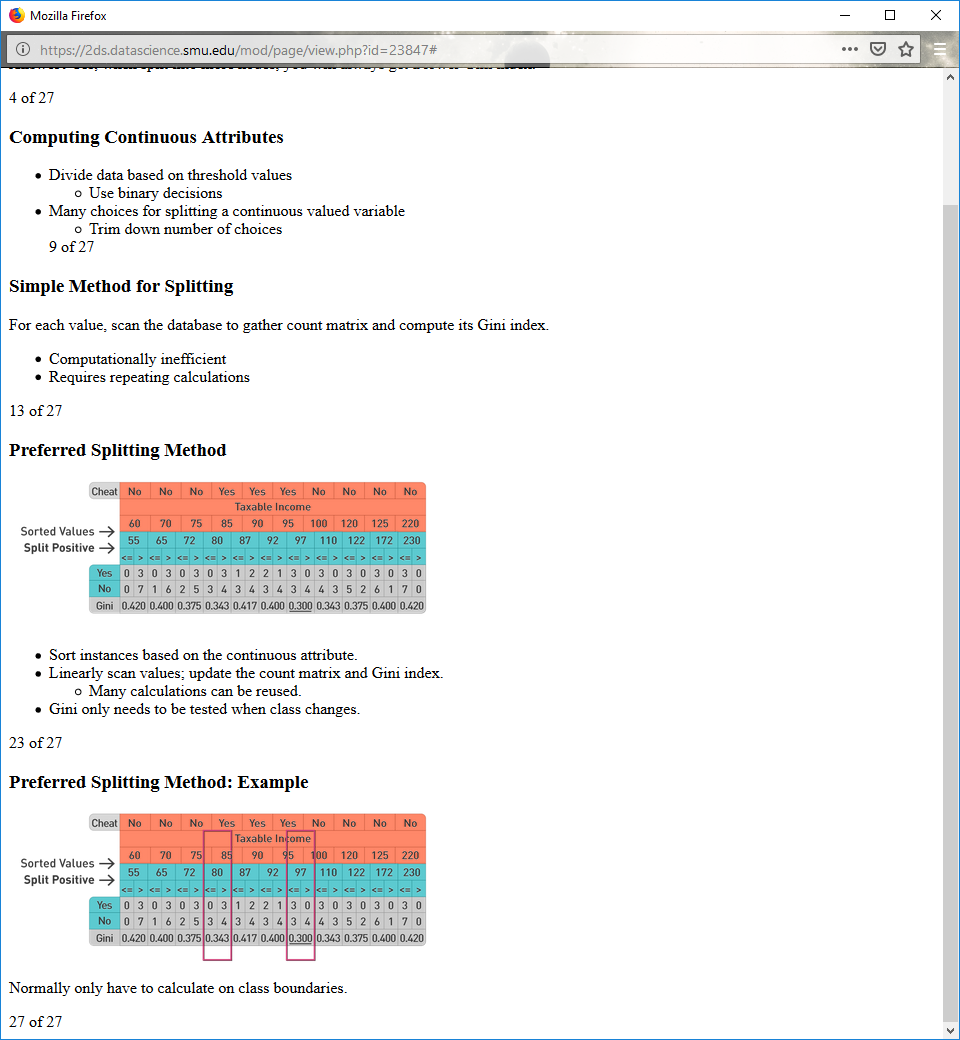
Sort instances based on the continuous attribute.

Linearly scan values; update the count matrix and Gini index.

Many calculations can be reused.

Gini only needs to be tested when class changes.

**Preferred Splitting Method: Example**



Normally only have to calculate on class boundaries.

**Entropy**

Information theoretic value

Comes from the problem of selecting how many bits to allocate to different classes

**Entropy Example: Text Compression**

Compressing document written in English

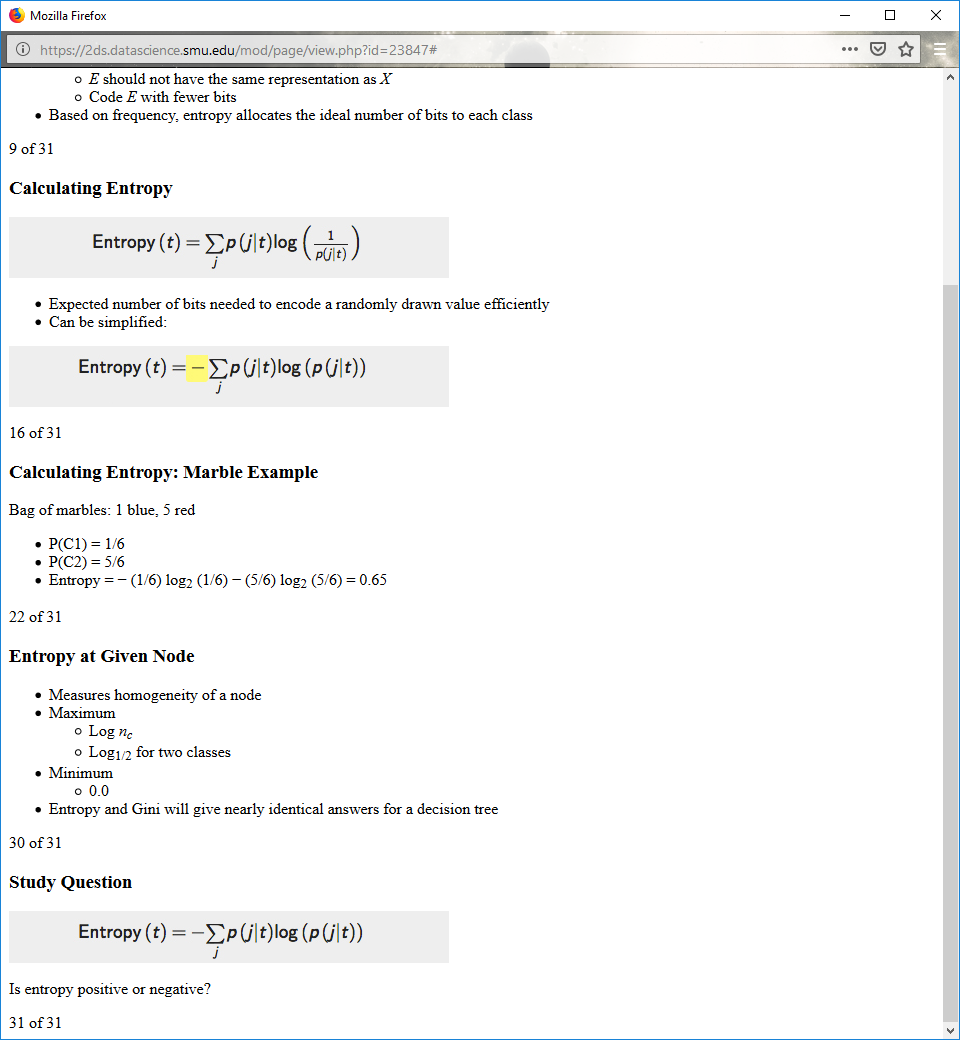
Most common letter is E

E should not have the same representation as X

Code E with fewer bits

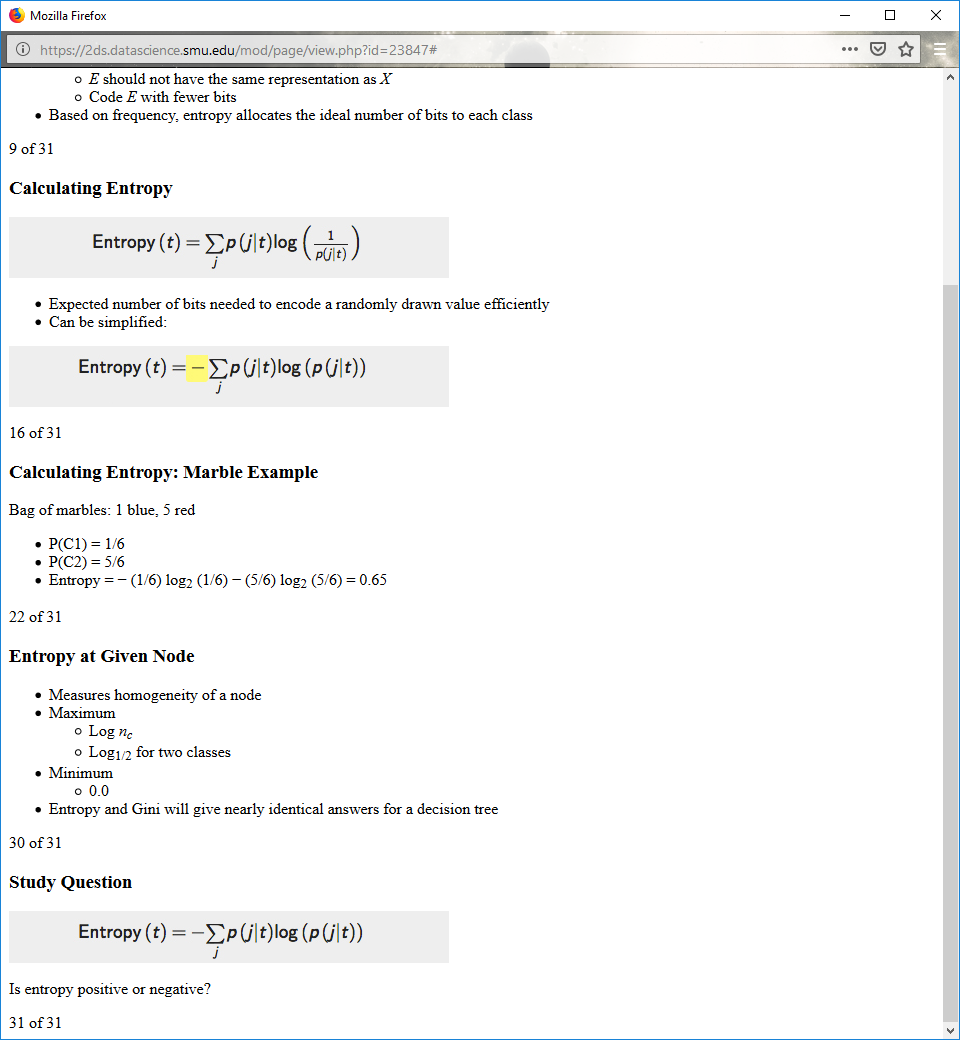
Based on frequency, entropy allocates the ideal number of bits to each class

**Calculating Entropy**



Expected number of bits needed to encode a randomly drawn value efficiently

Can be simplified:



**Calculating Entropy: Marble Example**

Bag of marbles: 1 blue, 5 red

P(C1) = 1/6

P(C2) = 5/6

Entropy = − (1/6) log2 (1/6) − (5/6) log2 (5/6) = 0.65

**Entropy at Given Node**

Measures homogeneity of a node

Maximum

Log nc

Log1/2 for two classes

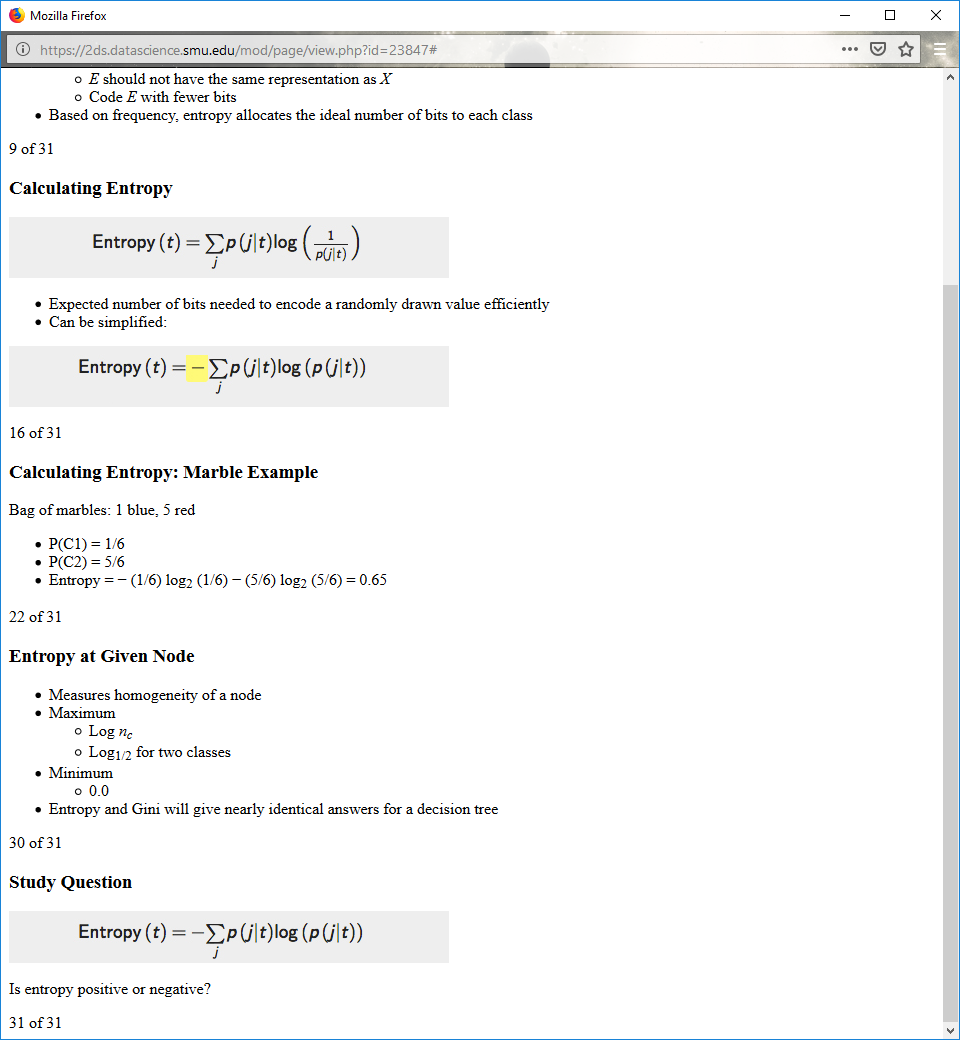
Minimum

0.0

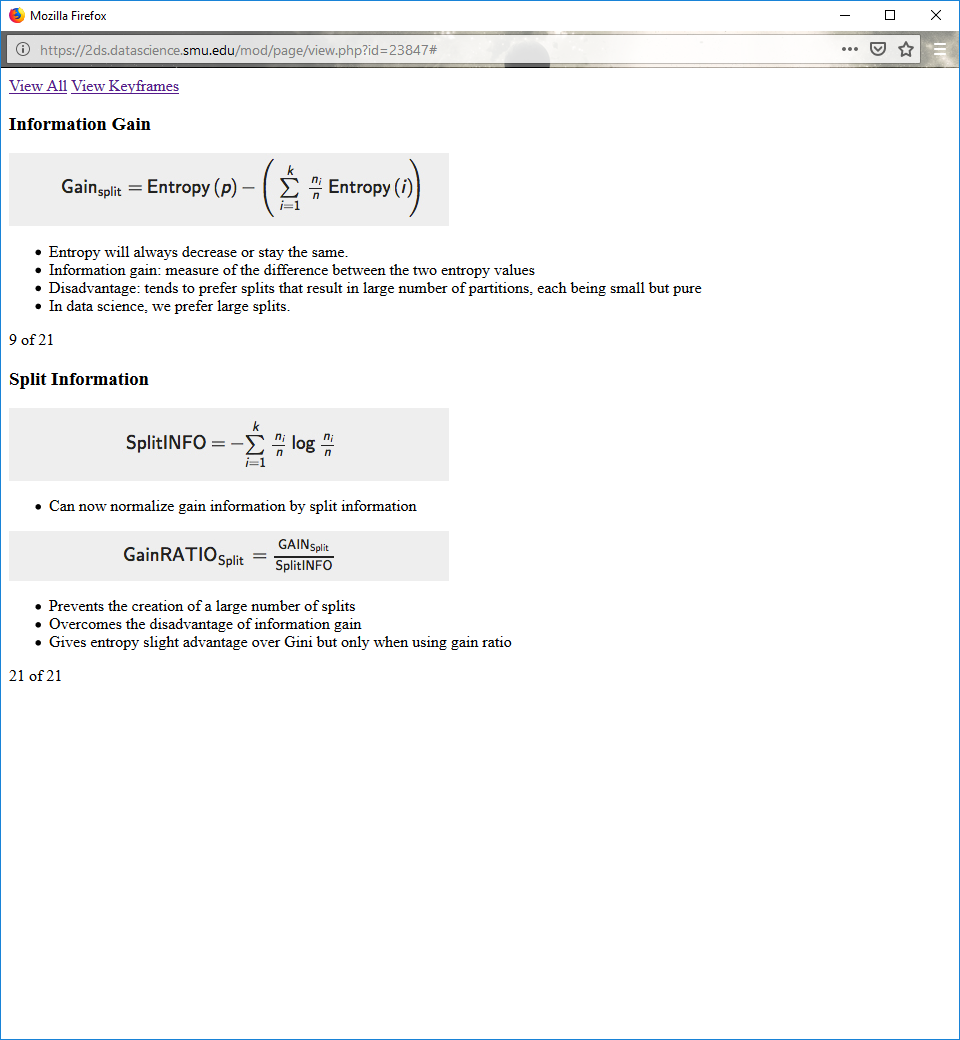
Entropy and Gini will give nearly identical answers for a decision tree

**Study Question**

Is entropy positive or negative?



**Information Gain**



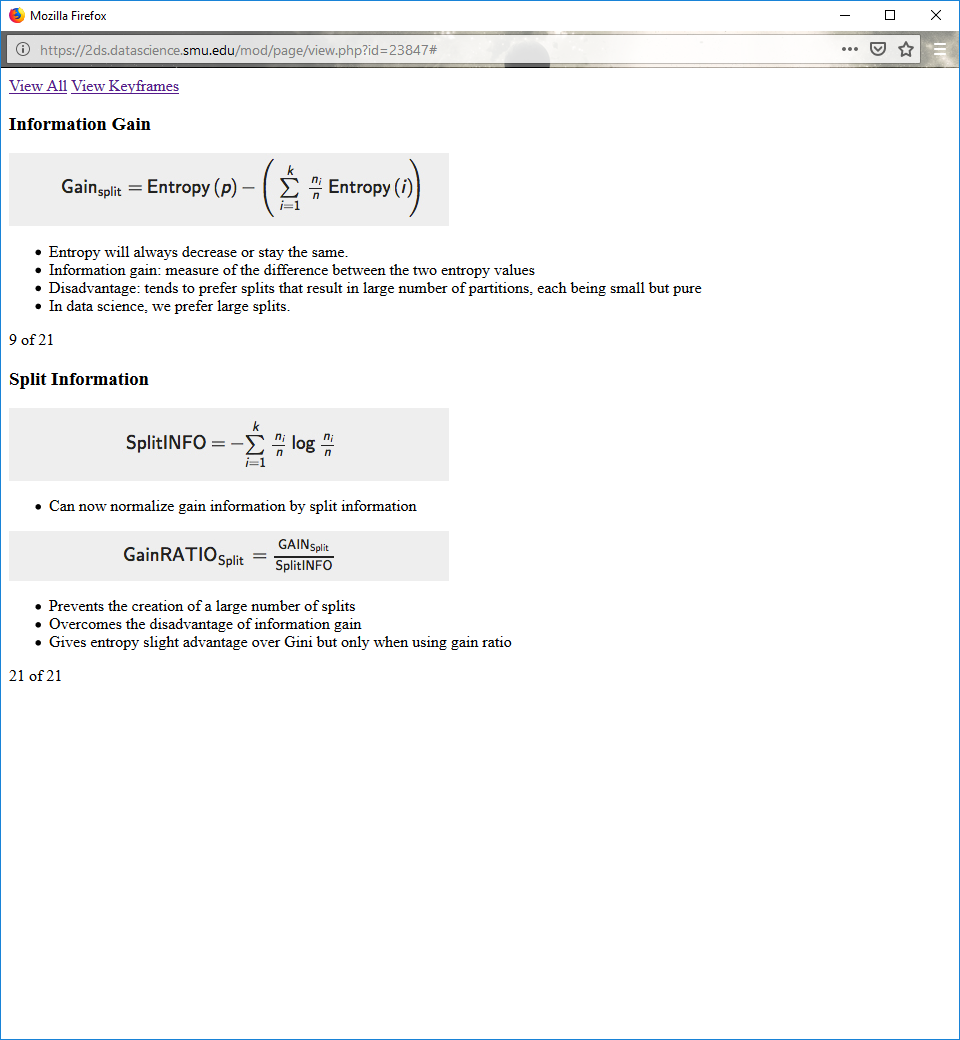
Entropy will always decrease or stay the same.

Information gain: measure of the difference between the two entropy values

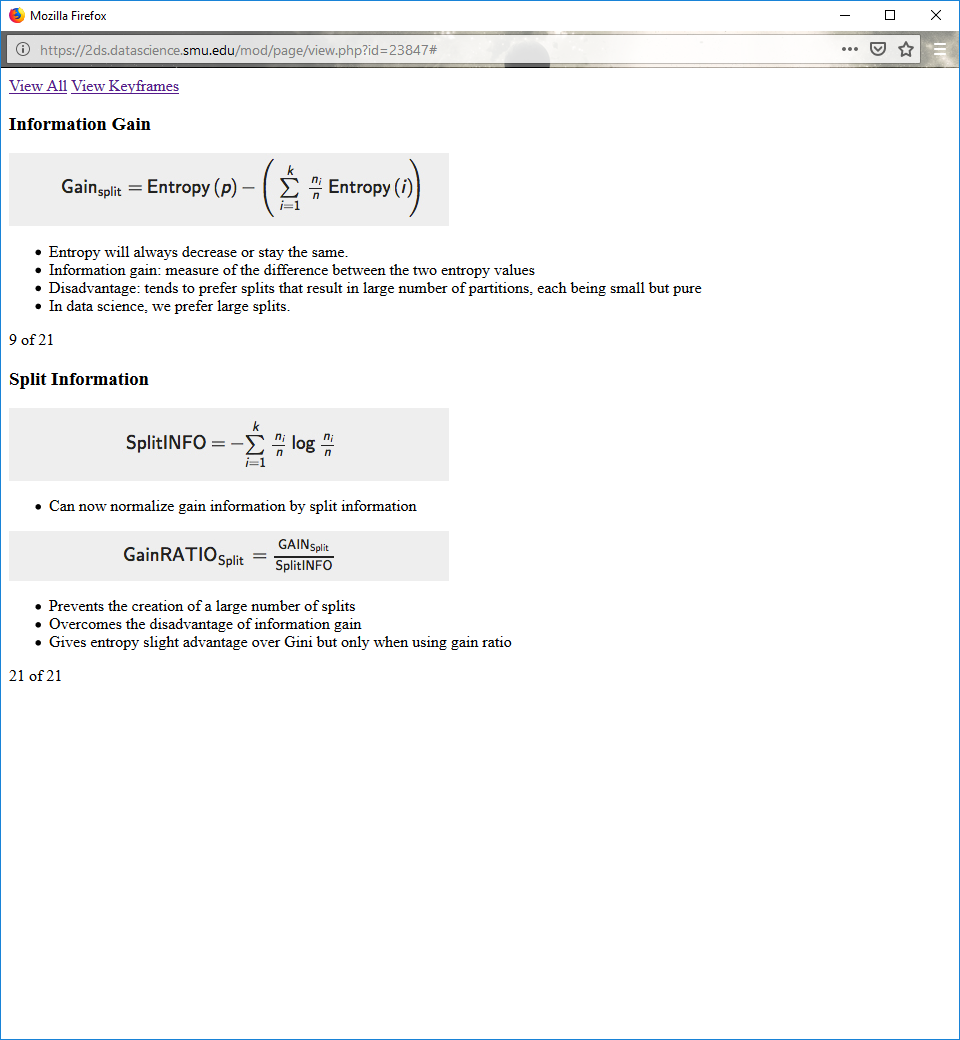
Disadvantage: tends to prefer splits that result in large number of partitions, each being small but pure

In data science, we prefer large splits.

**Split Information**



Can now normalize gain information by split information



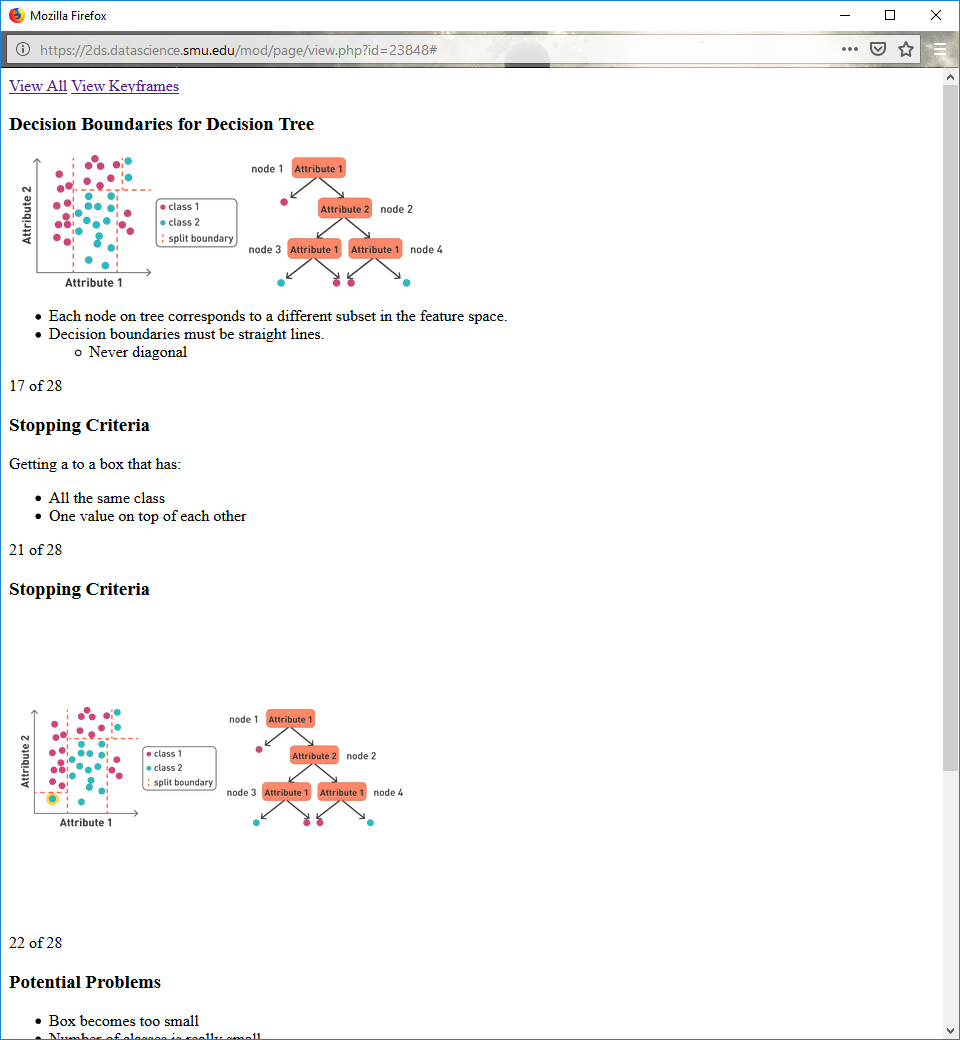
Prevents the creation of a large number of splits

Overcomes the disadvantage of information gain

Gives entropy slight advantage over Gini but only when using gain ratio

**6.6**

**Decision Boundaries for Decision Tree**



Each node on tree corresponds to a different subset in the feature space.

Decision boundaries must be straight lines.

Never diagonal

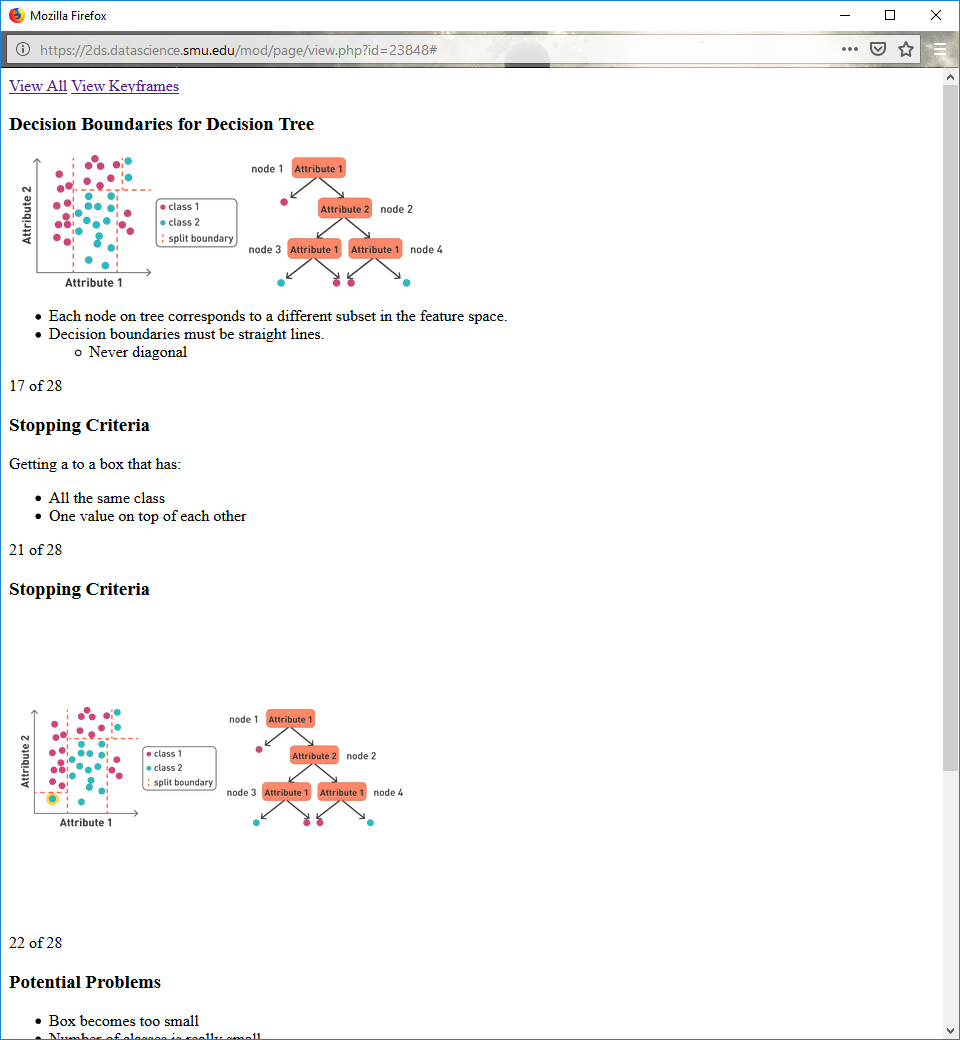
**Stopping Criteria**

Getting a to a box that has:

All the same class

One value on top of each other

**Stopping Criteria**



**Potential Problems**

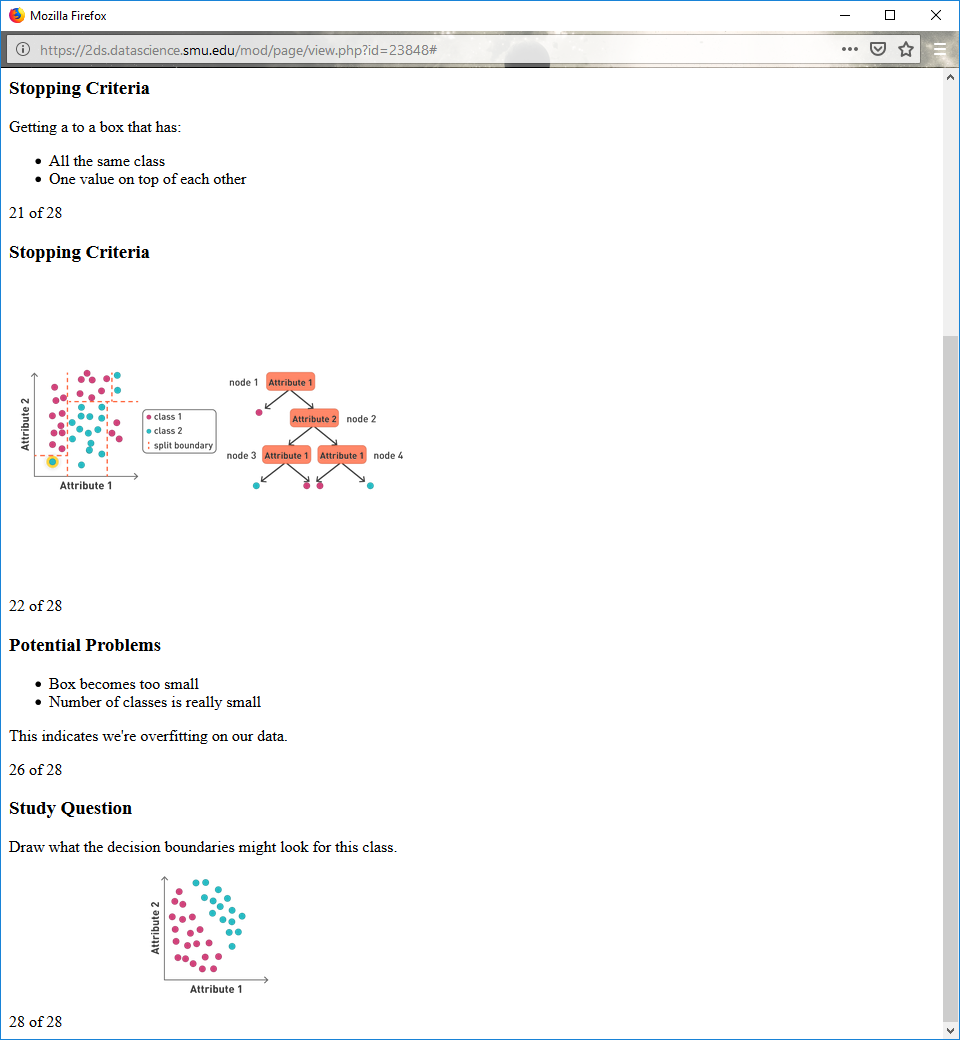
Box becomes too small

Number of classes is really small

This indicates we're overfitting on our data.

**Study Question**

Draw what the decision boundaries might look for this class.

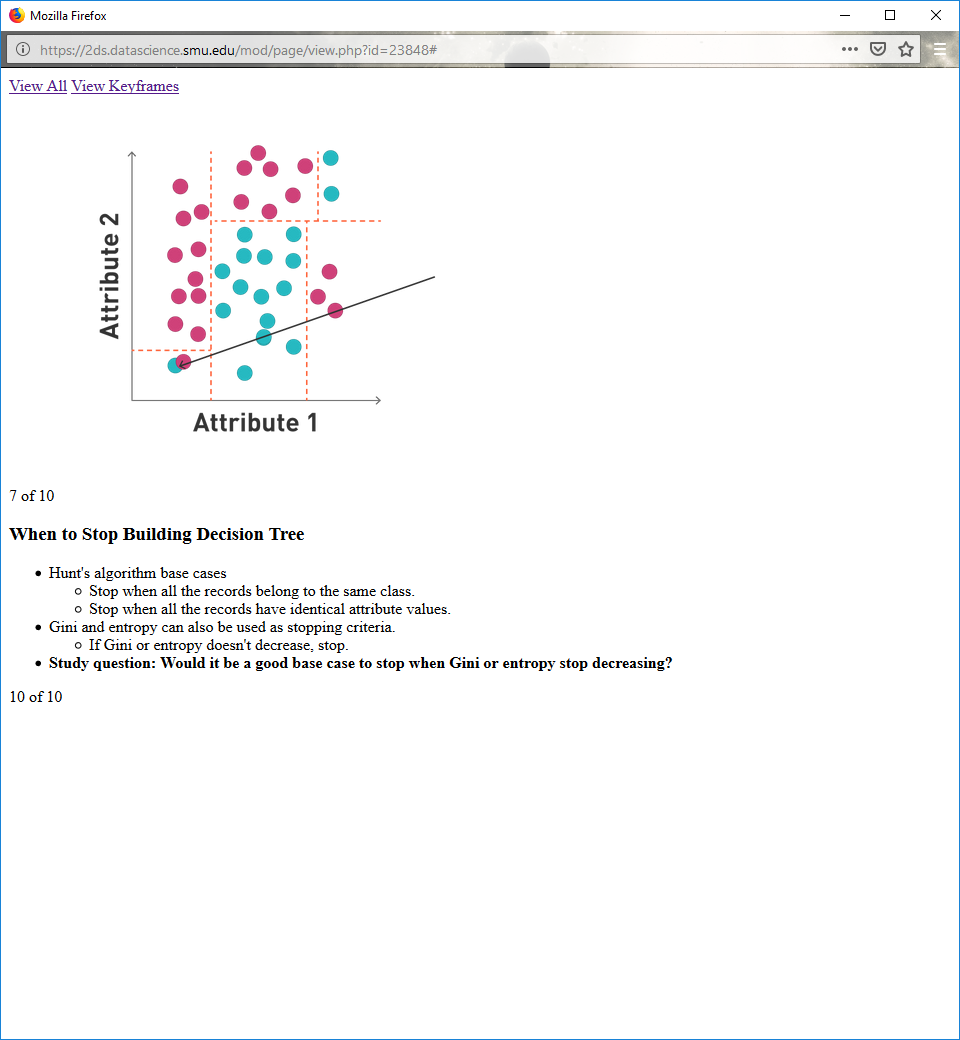


**When to Stop Building Decision Tree**

Hunt's algorithm base cases

Stop when all the records belong to the same class.

Stop when all the records have identical attribute values.



Gini and entropy can also be used as stopping criteria.

If Gini or entropy doesn't decrease, stop.

**Study question: Would it be a good base case to stop when Gini or entropy stop decreasing? (No?)**

**Common Decision Tree Algorithms**

ID3 (Iterative Dichotomizer 3)

Splits attributes in binary values

Uses information gain for binary splits

Requires ordinal or nominal data, making it difficult to use on continuous data

CART: Classification and Regression Tree

Uses Gini or Entropy

Looks for binary splits in data

Also breaks up continuous values according to the best split

**Common Decision Tree Algorithms: C4.5**

Java-based version called: J48

Uses gain ratio

Allows missing data (leaves missing values out of entropy calculation)

After creation, prunes tree to prevent overfitting

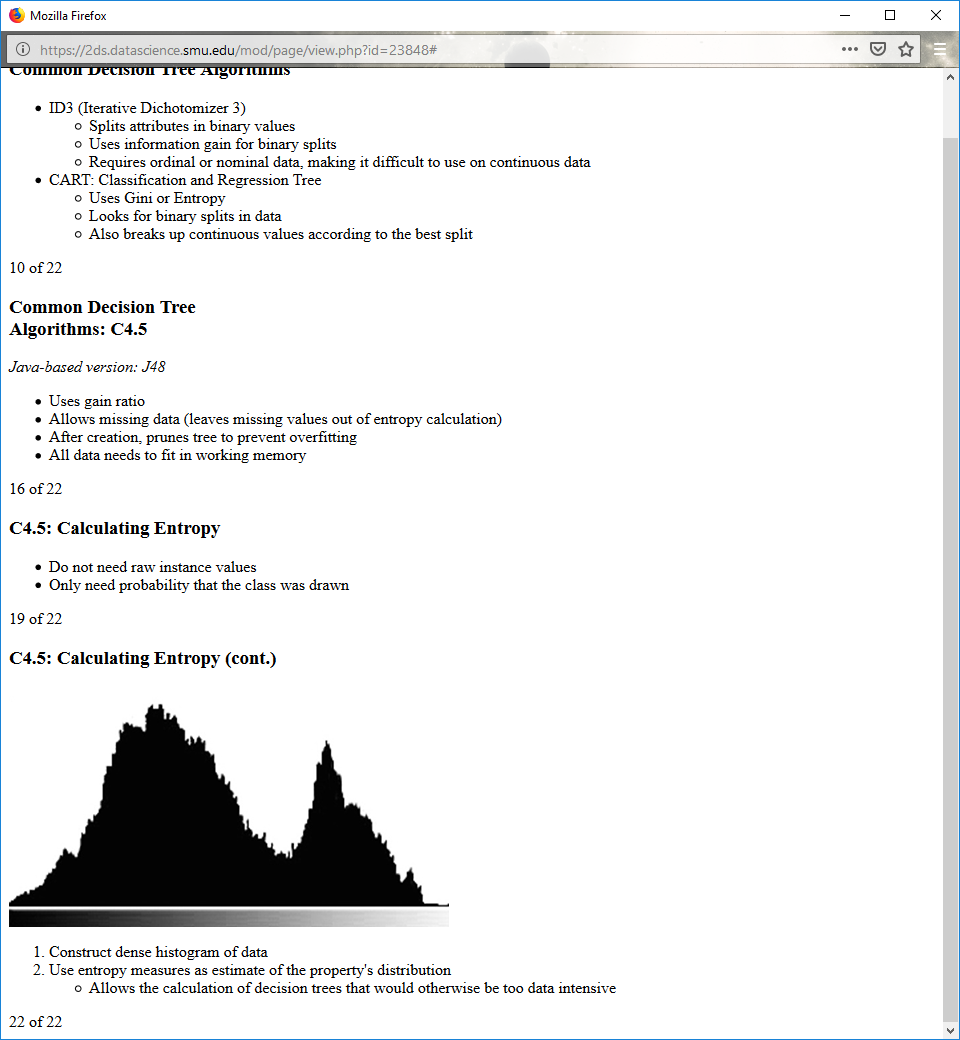
All data needs to fit in working memory

**C4.5: Calculating Entropy**

Do not need raw instance values

Only need probability that the class was drawn

**C4.5: Calculating Entropy (cont.)**



1. Construct dense histogram of data

2. Use entropy measures as estimate of the property's distribution

Allows the calculation of decision trees that would otherwise be too data intensive

**6.7**

**Decision Tree Overfitting**

Smaller boxes and fewer samples to make a selection from on how to split the data.

**Occam's Razor**

Given two models of similar generalization errors, which model is less complex?

Fewer branches or leaf nodes inside of tree

Smaller tree width

Prefer models with less nodes on them

**Choosing the Simpler Model Method 1: Pre-pruning**

Stop early.

If there are 10 examples, stop, choose majority class.

When difference between attributes is very small, stop, choose majority class.

**Choosing the Simpler Model Method 2: Post-pruning**

Grow decision tree to its entirety.

Trim the nodes of the decision tree in a bottom-up fashion.

Need an algorithm to decide whether or not to cut off branches

**Pessimistic Error**

Error of tree is not just the number of classes

C0: 9, C1: 1

Accuracy of the node is 90%, error in node is 10%

Add in number of leaf nodes (N)

Total errors: e(T) + N × 0.5

**Pessimistic Error: Example**

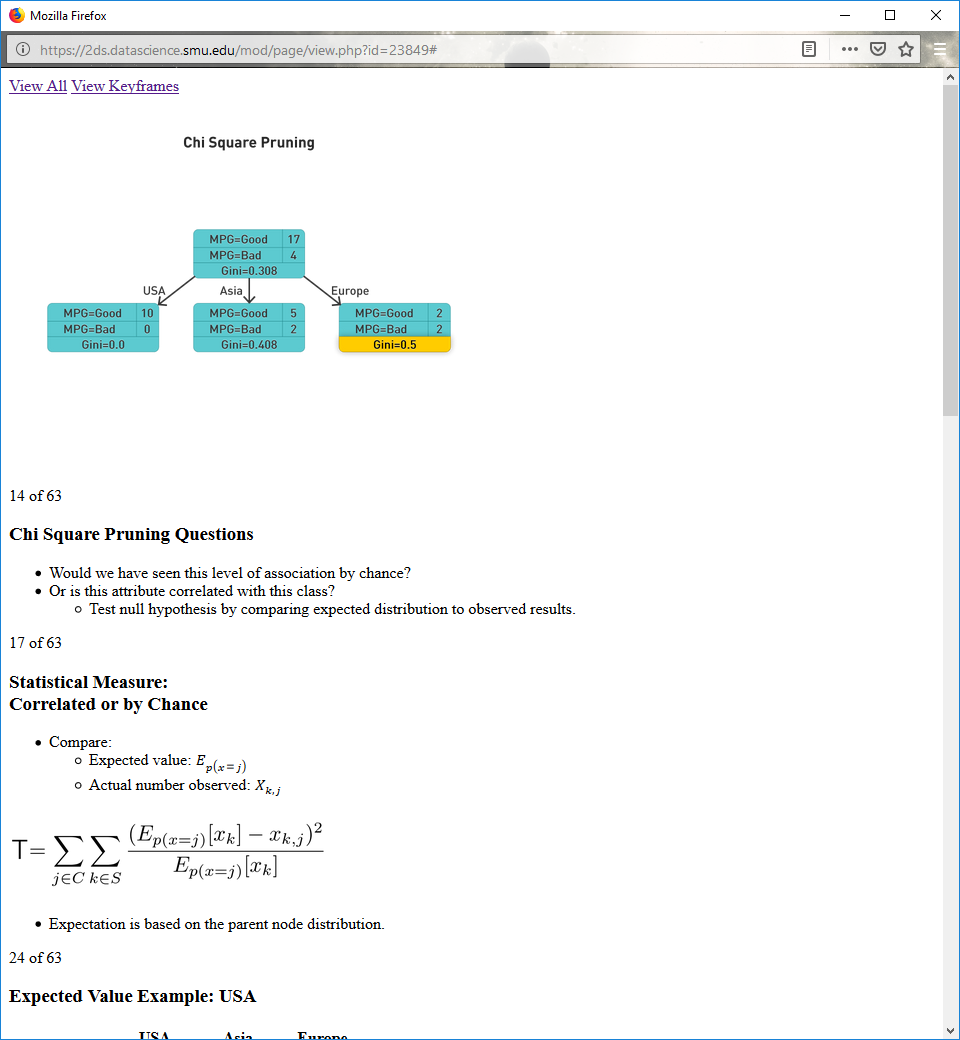
For a tree with 30 leaf nodes and 10 errors on training, out of 1000 instances:

Training error = 10/1000 = 1%

Penalize the complexity of the tree

Pessimistic error = (10 + 30 × 0.5)/1000 = 2.5%

**Chi Square Pruning**



**Chi Square Pruning Questions**

Would we have seen this level of association by chance?

Or is this attribute correlated with this class?

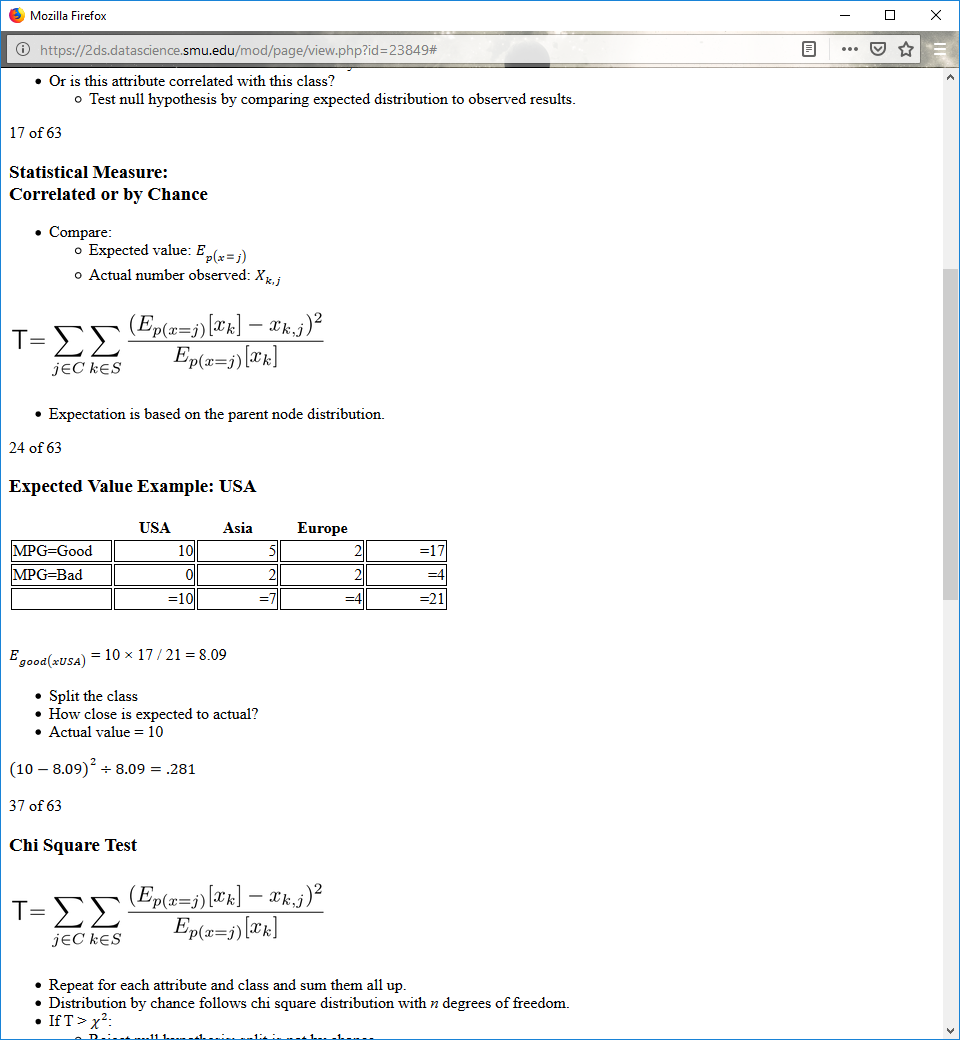
Test null hypothesis by comparing expected distribution to observed results.

**Statistical Measure: Correlated or by Chance**

Compare:

Expected value:

Actual number observed:



Expectation is based on the parent node distribution.

**Expected Value Example: USA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **USA** | **Asia** | **Europe** |  |
| MPG=Good | 10 | 5 | 2 | =17 |
| MPG=Bad | 0 | 2 | 2 | =4 |
|  | =10 | =7 | =4 | =21 |

= 10 × 17 / 21 = 8.09

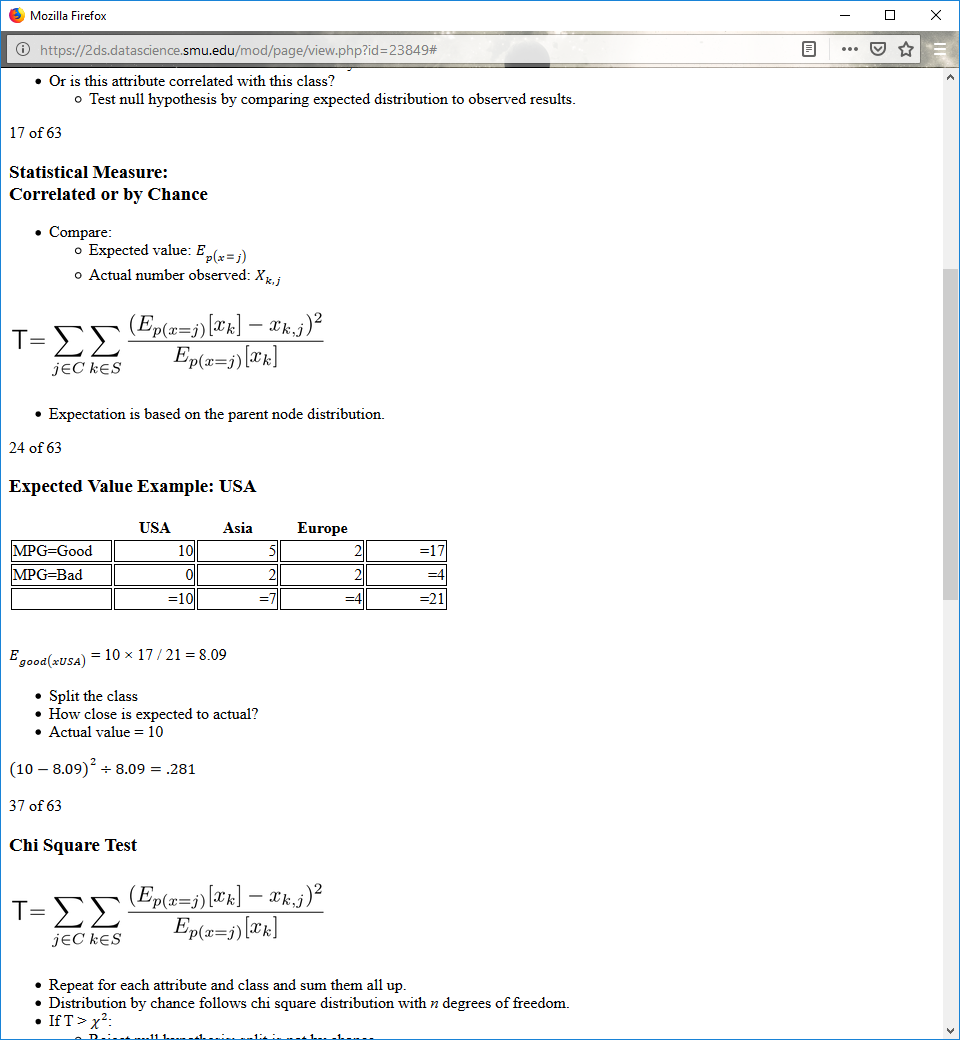
Split the class

How close is expected to actual?

Actual value = 10

( 10 − 8.09 ) 2 ÷ 8.09 = .281

**Chi Square Test**



Repeat for each attribute and class and sum them all up.

Distribution by chance follows chi square distribution with n degrees of freedom.

If T > χ2:

Reject null hypothesis; split is not by chance.

If T < χ2:

Accept null hypothesis; split is by chance.

**Chi Square Test: Example**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **USA** | **Asia** | **Europe** |
| MPG=Good | 10 (obs)  8.09 (exp) | 5 (obs)  5.66 (exp) | 2 (obs)  3.23 (exp) |
| MPG=Bad | 0 (obs)  1.91 (exp) | 2 (obs)  1.33 (exp) | 2 (obs)  0.76 (exp) |

= 10 × 17 / 21 = 8.09

MPGUSA: ( 10 − 8.09 ) 2 ÷ 8.09

MPGAsia: ( 5 − 5.66 ) 2 ÷ 5.66

Repeat for each set in the data and add them up

**Chi Square Test: Example (cont.)**

|  |  |
| --- | --- |
| **df** | **χ2 .050** |
| 1 | 3.841 |
| 2 | 5.991 |
| 3 | 7.815 |
| 4 | 9.488 |
| 5 | 11.070 |

Goal: 95% confidence the split is not by chance

df = (rows − 1) × (cols − 1)

df = (2 − 1) × (3 − 1) = 2

For 95% confidence, sum must be greater than 5.991

Squared values = 5.25; < 5.991

Cannot say it is a good split

**Study Question**

Should you prune the tree based on chi square statistic and pessimistic error?

|  |  |
| --- | --- |
| Class = Yes | 20 |
| Class = No | 10 |
| Error = 10/30 | |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **A1** | | **A2** | | **A3** | | **A4** | |
| Class = Yes | 8 | Class = Yes | 3 | Class = Yes | 4 | Class = Yes | 5 |
| Class = No | 4 | Class = No | 4 | Class = No | 1 | Class = No | 1 |

**Advantages of Decision Trees**

Inexpensive to construct

Extremely fast at classifying unknown records

Easy to interpret for small-sized trees

Accuracy is comparable to other classification techniques for many data sets