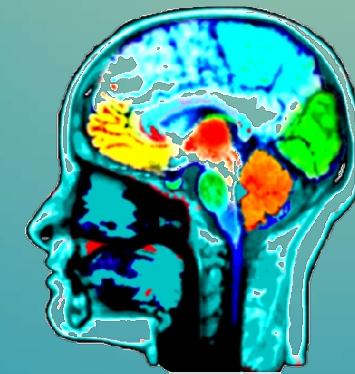




Introduction To Data Mining

Isfahan University of Technology (IUT)
Bahman 1401



Classification

Dr. Hamidreza Hakim
hamid.hakim.u@gmail.com

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

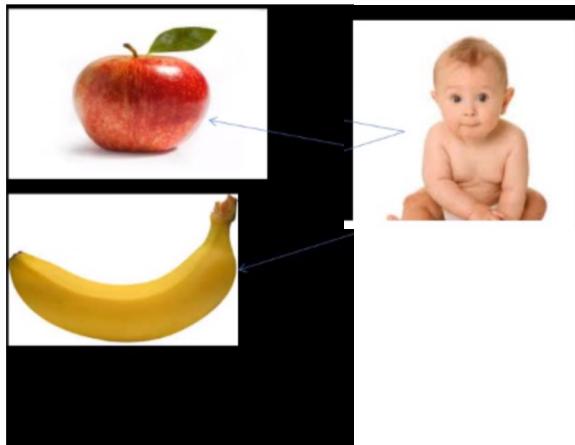
Content

Classification: Basic Concepts

Decision Tree Induction

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
- Unsupervised learning (clustering)



<https://gowthamy.medium.com/machine-learning-supervised-learning-vs-unsupervised-learning-f1658e12a780>



<https://www.mehrnews.com/news/4969404>

-
دو نوع یادگیری داریم:
یادگیری با نظارت:

مثلًا به یک کودک والدینش بهش می‌گن این سبب است یا .. و این کودک هم این پدیده رو می‌بینه و
هم برچسبی که قراره یاد بگیره رو می‌بینه
یادگیری بدون نظارت:

هیچ اطلاعاتی راجع به برچسب نداریم و به صورت صریح بهمون نمی‌گن دنبال چه حقیقتی هستیم
مثلًا دعوت شدیم به یک مهمونی و به ما می‌گن که می‌تونیم بگیم فامیل های عروس و داماد کیا
هستن --> با یک نگاه به خود تالار می‌بینیم ادمایی که با هم فامیل هستن نزدیک هم قرار گرفته اند
و ما به عنوان یک ناظر بیرونی بدون اینکه بدونیم کی برچسبش چی هست می‌تونیم یکسری
اطلاعاتی به دست بیاریم

Supervised vs. Unsupervised Learning

- Supervised learning (**classification**)
 - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
 - New data is classified based on the training set
- Unsupervised learning (**clustering**)
 - The class labels of training data is unknown
 - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

-
توی یادگیری با نظارت یکسری دیتا داریم تحت عنوان داده های اموزشی یعنی نمونه هایی که مشاهده و اندازه گیریشون کردیم و کنار اون داده ها یک برچسبی داریم تحت عنوان **label** که بهمون میگه این داده هایی که مشاهده میکنیم قراره چه مقدار یا برچسبی رو داشته باشن هدف توی یادگیری با نظارت اینه که یک دیتایی به ما میدن و ما نمی دونیم برچسبش چیه و نمی دونیم این چه دسته ای هست و قراره با اطلاعاتی که توی داده های اموزشی هست یک برچسب مناسب برای این داده های جدید پیدا بکنیم
این برچسب ها رو عمدتا انسان ها استخراج می کنن یعنی به صورت انسانی اینا تولید شده و میدونیم این برچسب ها به چه دسته ای تعلق دارن

توی یادگیری بدون نظارت خبری از برچسب نیست و برچسب ها ناشناخته هستن

Problems: Classification vs. Prediction

- Classification
 - predicts categorical class labels (discrete or nominal)
 - classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data
- Prediction
 - models continuous-valued functions, i.e., predicts unknown or missing values

یادگیری با ناظارت به دو دسته تقسیم میشه:
Prediction و Classification

و فرق این دوتا توی اون نوع برچسب هست --> اگر برچسب ها به صورت گسته باشن و مقادیر متمایزی داشته باشن ما میگیم **class labels** داریم یعنی یک برچسب کلاسی داریم و یک مسئله **Prediction** طبقه بندی است ولی اگر مقادیر پیوسته باشن مسئله میشه

Prediction Problems: Classification vs. Prediction

- Typical applications
 - Credit/loan approval:
 - Medical diagnosis: if a tumor is cancerous or benign
 - Fraud detection: if a transaction is fraudulent
 - Web page categorization: which category it is

Examples of Classification Task

Task	Attribute set, x	Class label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from x-rays or MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

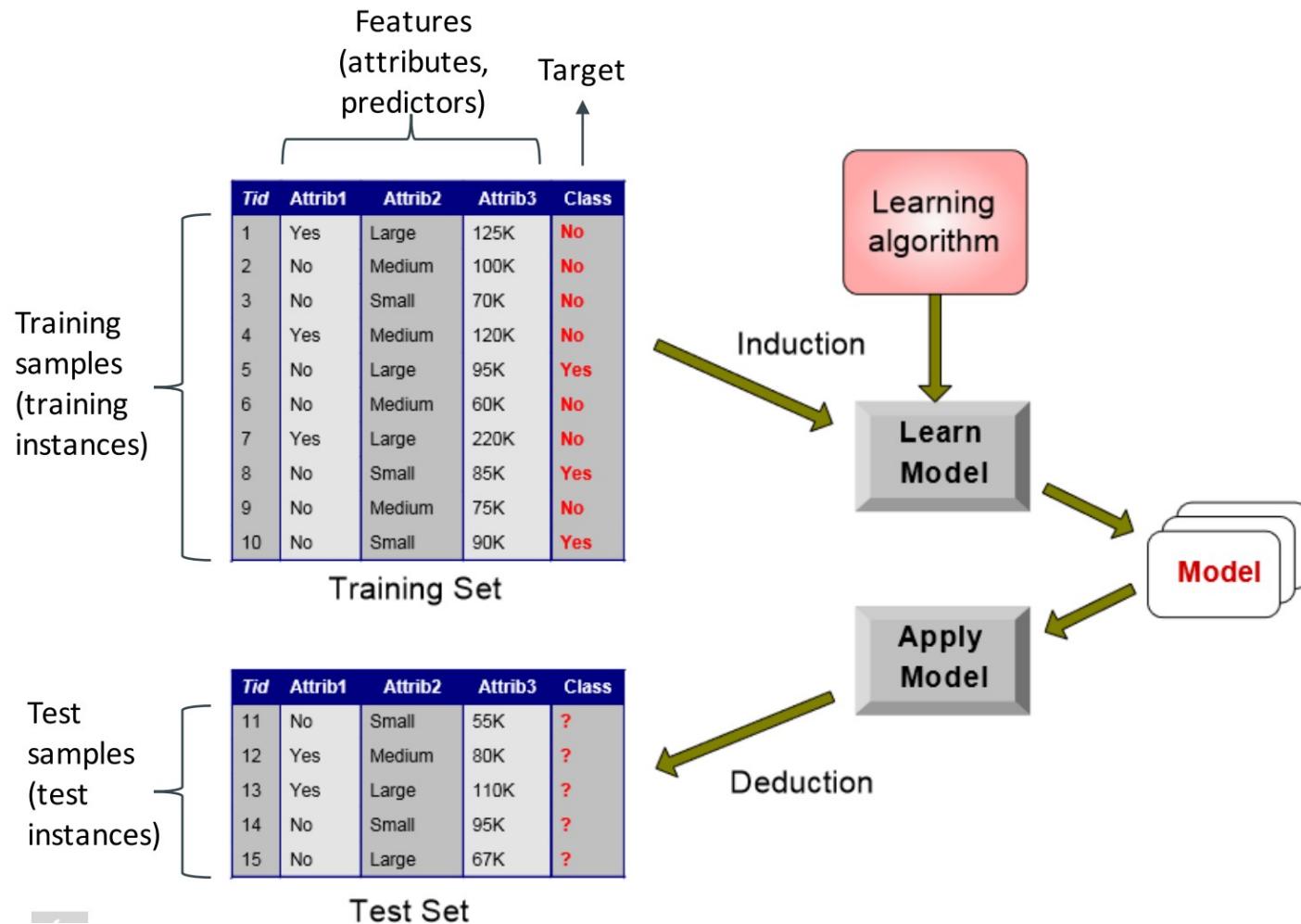
1- هدف: دسته بندی ایمیل ها به دو دسته اسپم و غیر اسپم --> اینجا یه چیزی داریم تحت عنوان Attribute و Attribute بینی اون اطلاعاتی که ما راجع به پدیده داریم که اینجا ایمیل ها هستن و Attribute هایی که روی ایمیل می تونیم داشته باشیم: کلماتی که داخلش هست - مبدا که ایمیل ارسال شده - محتوای خود ایمیل

2- برای شناسایی سلول های سرطانی: ما یک تصویر MRI یا x-ray گرفتیم و میخوایم بگیم این سلول سرطانی هست یا نه --> میزان تصویر که اون سلول داره می تونه یک Attribute باشه و برچسب این باشه که بگیم این سلول مشکوک هست یا نه و چجوری این برچسب ها رو جمع اوری می کن؟ یکسری کارشناس روی یک تعداد تصویر برامون این کارو می کن

:3

فهرست نویسی کهکشان ها
ویژگی های استخراج شده از تصاویر تلسکوپ
کهکشان های بیضوی، مارپیچی یا نامنظم

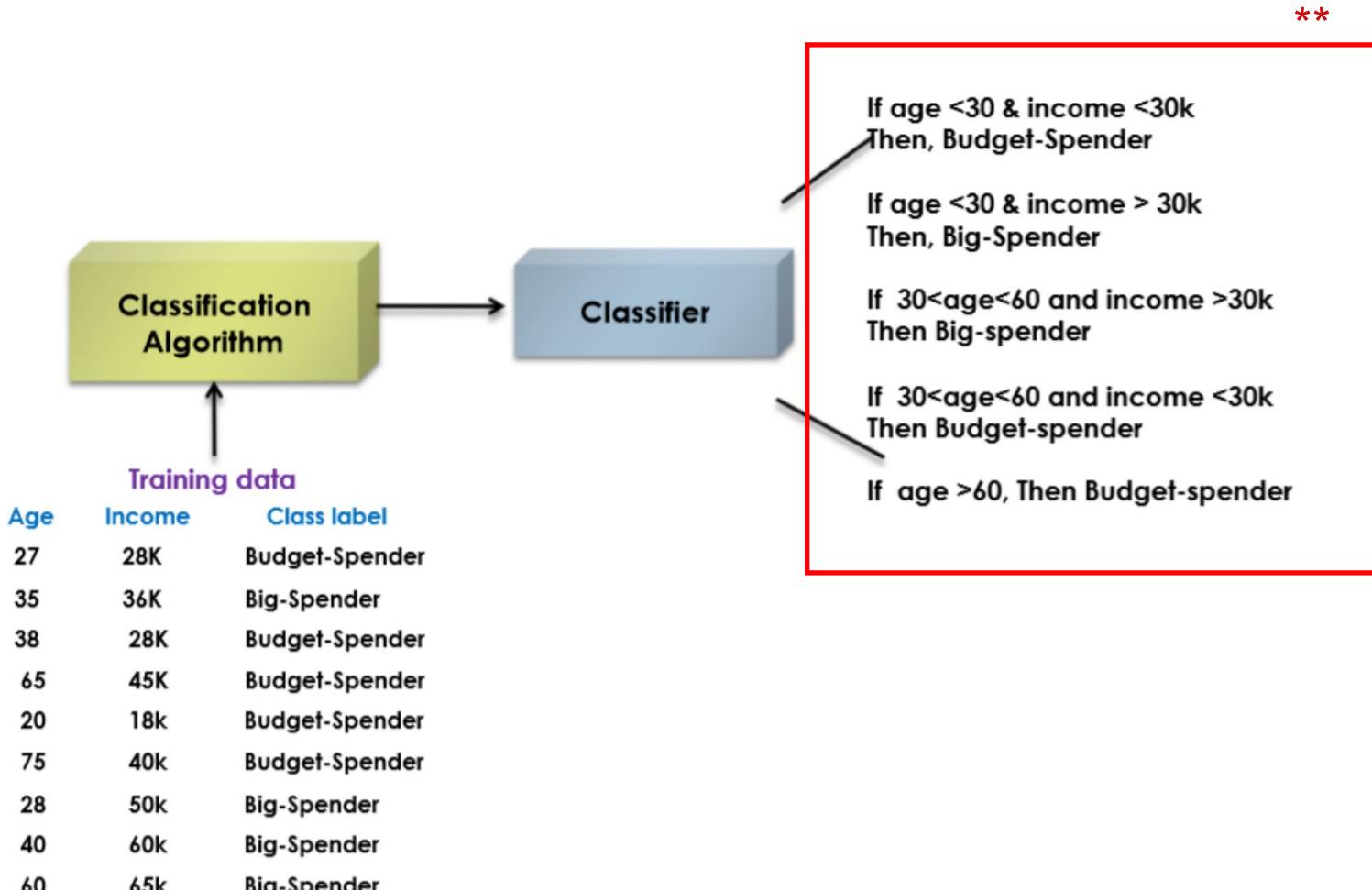
Classification Process



توی طبقه بندی اول از همه داده های اموزشی داریم که بهش می گن training samples نمونه های اموزشی --> توی این نمونه های اموزشی ویژگی ها کاملا مشخص هستن و کلاس هم مشخصه و به این ویژگی ها features يا .. هم میگن این نمونه های اموزشی رو میدیم دست یک مدل یاد گیرنده ای یعنی learning model یادگیرنده با کمک یک الگوریتم یادگیری سعی میکنه الگویی که توی این داده ها وجود دارن رو استخراج بکنه و اون الگو رو که استخراج کرد اونو تحت قالب یک مدلی ذخیره بکنه و یک مدلی بهمون بده

بعد که این مدله ایجاد شد ما میریم سراغ استفاده از یک Classifier که مرحله تستش هست و توی تست یکسری نمونه داریم که این نمونه ها برچسب ندارن و این عملیات رو انجام دادیم که برچسب اینارو مشخص بکنیم سعی میکنیم این test samples رو یعنی تک تک این رکوردها رو بدیم به اون مدل و طی یک فرایندی که بهش میگیم apply model یا به کارگیری مدل میایم نظر مدل رو راجع به اون رکورد می بینیم که چی هست

Training phase , Model construction



داده های اموزشی

توى طبقة بندى چنلتا فاز داريم:

1- فاز اموزش:

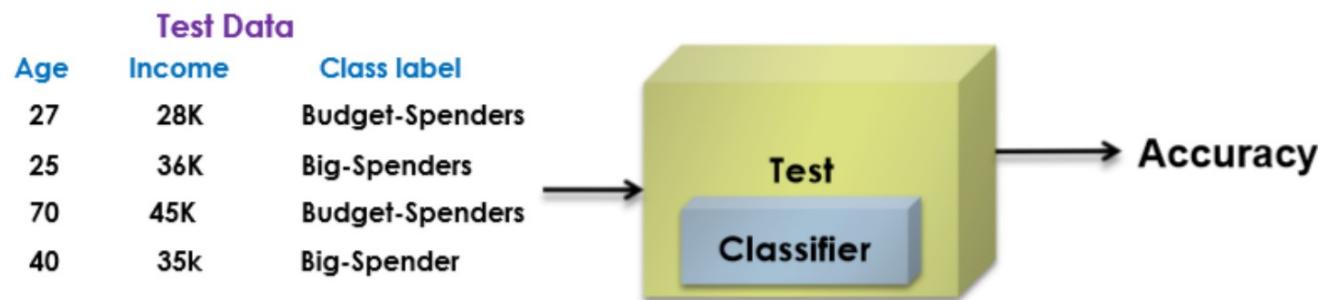
اين فاز همين فرایнд استخراج الگو از داده ها است --> میخوايم داده های اموزشی رو تبدیل بکنیم به يك مدلی--> اين داده ها اگر بخود با کمک Classifier ترین بشه تبدیل میشه به همچین ** قانون هایی

وقتی اين قانون ها شکل گرفت می ریم سراغ يك فازی به اسم فاز Evaluation يا فاز استفاده از مدل

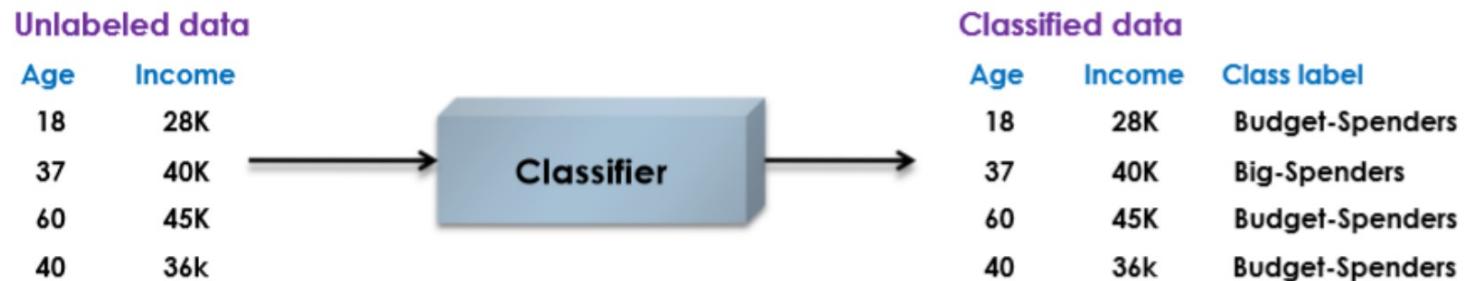
2- فاز Evaluation

Evaluation and usage phases

1-Test the classifier



2-If acceptable accuracy



فاز Evaluation

این فاز Evaluation یک گامی است که به ما می‌گوید مدلی که یاد گرفتیم چقدر خوب بوده

بعد از این که مدلی استخراج شد باید برایم تستش بکنیم --> تست جوری است؟ یکسری دیتاهايی که خودمون می‌دونیم برچسبش چیه ولی قبل از توی داده های اموزشی این ها رو نیاوردیم اینها رو مشخص می‌کنیم که اینا تحت عنوان test data ما هستن و اینها رو میدیم به مدلی که ایجاد کردیم و ازش می‌خوایم نظر بده و نظراتش که مشخص شد ممکنه نظراتش با اینها مطابقت داشته باشه و ممکنه یکسری جاهای هم نباشه در نهایت می‌ایم نتیجه اش رو با کمک یکسری معیارهای ارزیابی گزارش می‌کنیم مثل می‌ایم دقت رو گزارش می‌کنیم

نکته: اگر به اون سطح دقیقی که مورد نظرمان بود رسید می‌گیم این مدلمن، مدل خوبی است

Evaluation and usage phases

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}.$$

Table 3.4. Confusion matrix for a binary classification problem.

		Predicted Class	
		Class = 1	Class = 0
Actual Class	Class = 1	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

یک معیار ارزیابی از این که یک Classifier چقدر خوبه معیار ارزیابی accuracy است
دقت: تعداد پیش بینی های درست / تعداد کل پیش بینی ها

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
 - The set of tuples used for model construction is **training set**
 - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
 - Estimate accuracy of the model
 - ◆ The known label of test sample is compared with the classified result from the model
 - ◆ **Accuracy** rate is the percentage of test set samples that are correctly classified by the model
 - ◆ **Test set** is independent of training set (otherwise overfitting)
 - If the accuracy is acceptable, use the model to **classify new data**
- Note: If *the test set* is used to select models, it is called **validation (test) set**

- فرایند Classification: ما دو تا گام داریم یک گام ساخت مدل که داده ها رو به مون میدن و برچسب هاش هم مشخصه و میگن یک مدل ایجاد بکن و گام دوم استفاده از مدل است که توی این گام مهمترین بخشش این است که ما بگیم وقت چقدر بوده و معیار ارزیابی مناسب برای مدلمنون چی هست و مطمئن بشیم که مدلمنون به خوبی داده ها رو یاد گرفته یا نه

Binary classification vs Multiclass classification

Binary classification

- Classification tasks with only two classes, typically denoted by $\{+,-\}$, $\{+1,-1\}$, or $\{\text{Pos}, \text{Neg}\}$.
- Example: email spam detection, (pos/neg) sentiment analysis.

Multiclass classification

- Classification tasks with more than two classes.
- Example: email topic detection, (pos/null/neg) sentiment analysis.

دوتا مدل classification داریم:

Binary classification: ینی کلاس ما باینری است ینی دو کلاسه است --> وقتی کلاس ها دو حالته باشد مثلا مثبت باشه یا منفی باشه یا توی تشخیص بیماری مثلا دارد یا ندارد ...

Multiclass classification: ینی چند کلاسه است --> یکسری تکنیک وجود داره که این چند کلاس را تبدیل میکنه به دو کلاس

نوع classification هم توی انتخاب classifier یکسری جنبه ها رو برامون مهم می کنه و هم توی محصول نهایی

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Nearest-neighbor
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines
- Neural Networks, Deep Neural Nets
- Ensemble Classifiers
 - ◆ Boosting, Bagging, Random Forests
- And many more

DECISION TREE

درخت تصمیم

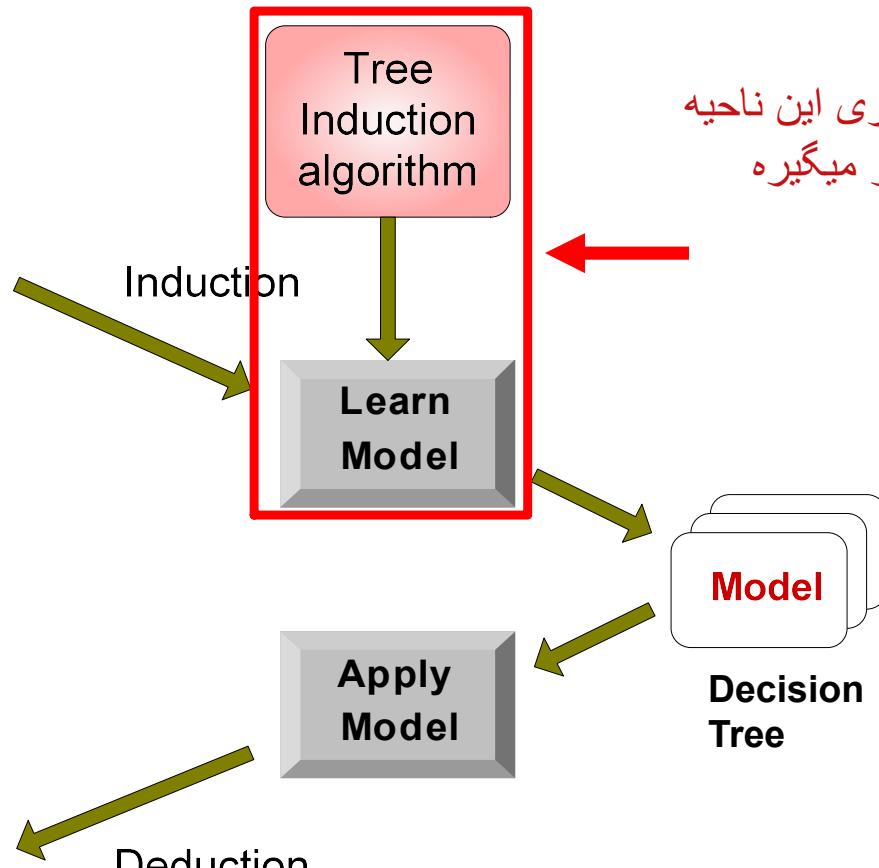
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



درخت تصمیم توی این ناحیه
قرمز رنگ قرار میگیره

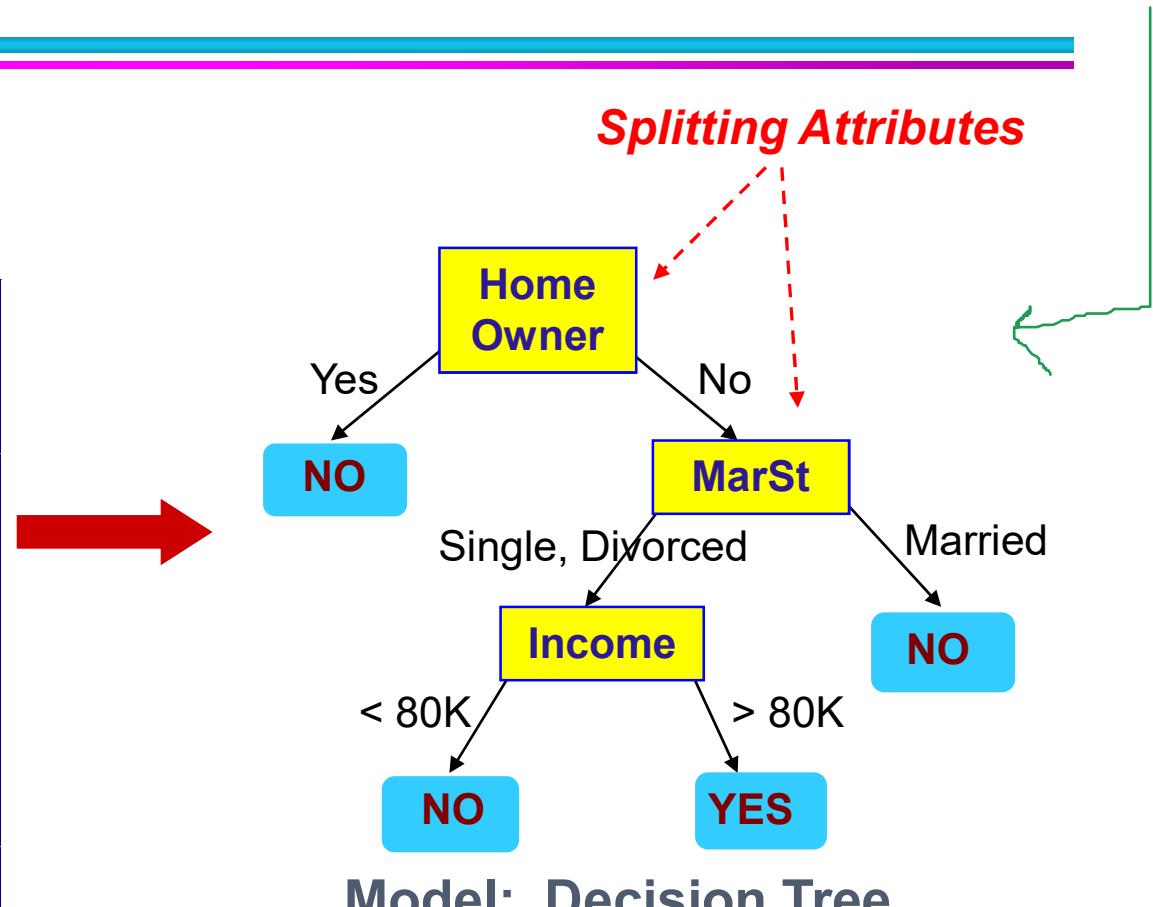
مدلی که اینجا استخراج میکنه و ایجاد میشه مدل درختی است ینی مدلش تبدیل میشه به یک درخت تصمیم

این یک مدل است که می توانه معرفی
باشه برای کل این داده ها:

Example of a Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



- Internal nodes (non-leaf nodes) denote a test on an attribute.
- Branches represent outcomes of tests.
- Leaf nodes (terminal nodes) hold class labels.
- Root node is the topmost node.

مثال:

مسئله ای داریم و اون این است که ما جای یک کارشناس بانک هستیم و افراد اومدن یک حسابی رو باز کردن و این اطلاعات رو از خود افراد گرفتیم

مدل درخت تصمیم از چه اجزایی تشکیل شده؟

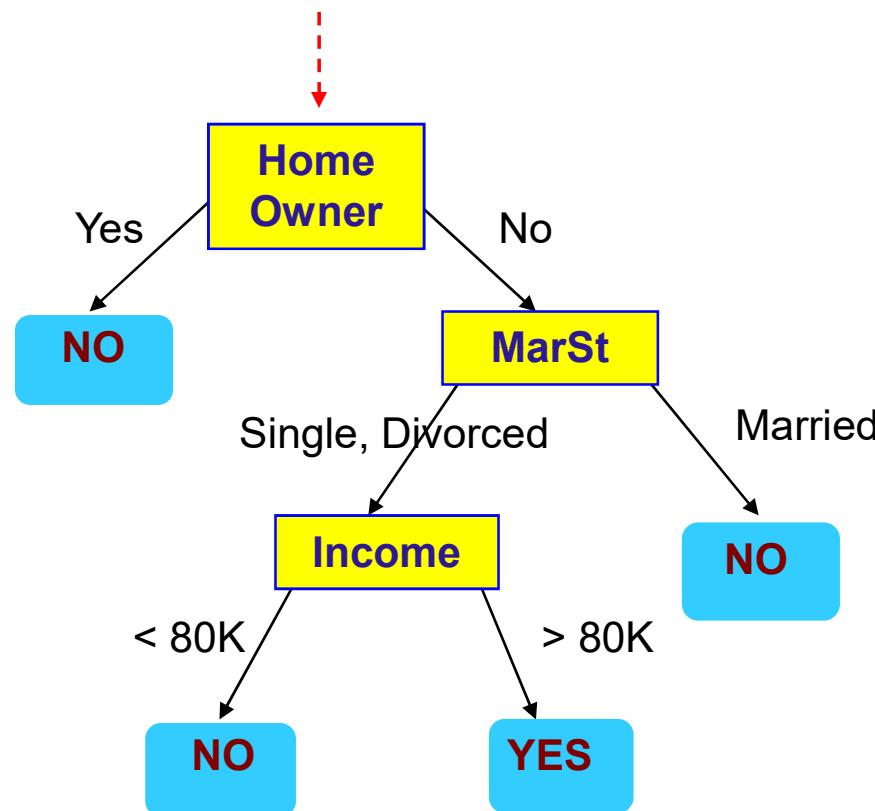
یکسری نود داریم که بهش میگیم نودهای داخلی --> رنگ های زرد --> این نودها به نوعی همین ویژگی هایی ما هستن که توی جدول اشاره شده

به واسطه هر نود داخلی ما یکسری برنچ داریم یا شاخه داریم که این ویژگی اگر یک مقداری رو داشت براساس مقادیری که اون ویژگی داره می ره روی یک سمت شاخه ها
این درخت یکسری برگ هم داره --> رنگ های ابی

Apply Model to Test Data

Test Data

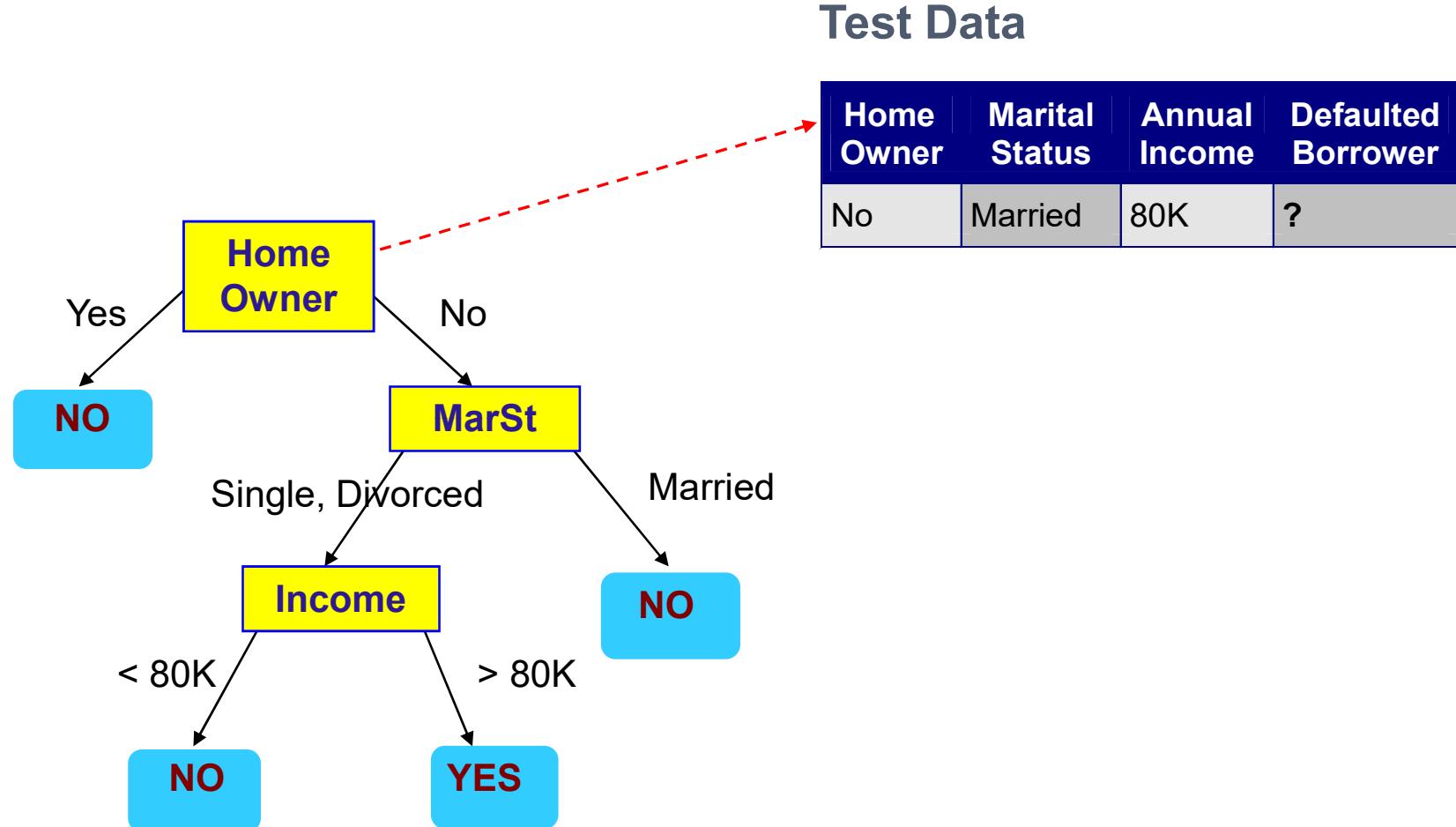
Start from the root of tree.



Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

یک دیتایی به ما میدن به اسم تست دیتا و میگن براساس این مدل (درخت) بگو برچسبش چی میشه

Apply Model to Test Data

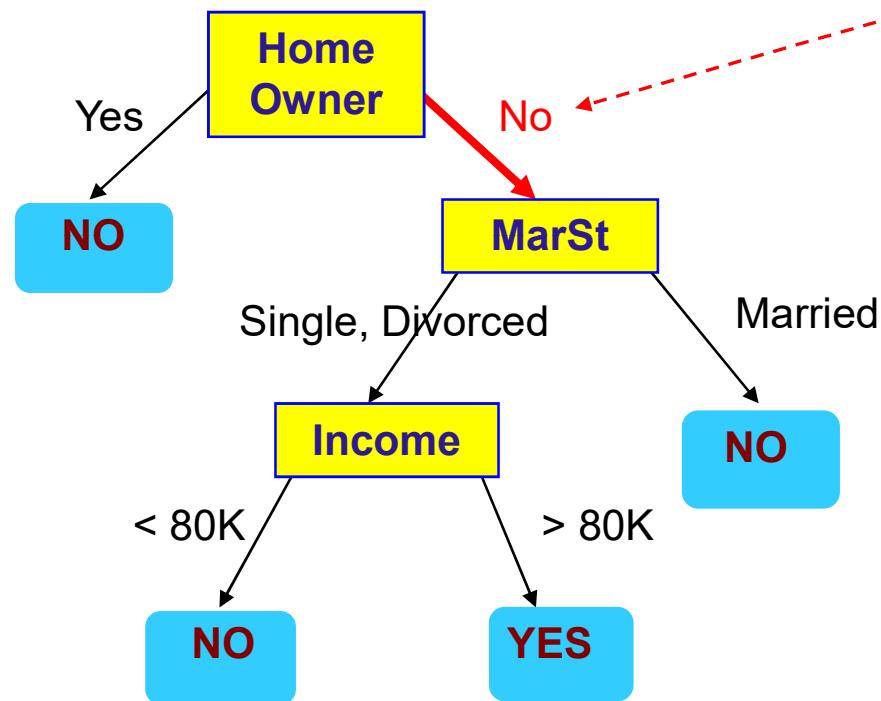


از ریشه شروع میکنیم بررسی میکنیم و می خونیم و می ریم به سمت پایین توی درخت تا به اون
برچسب بررسیم

Apply Model to Test Data

Test Data

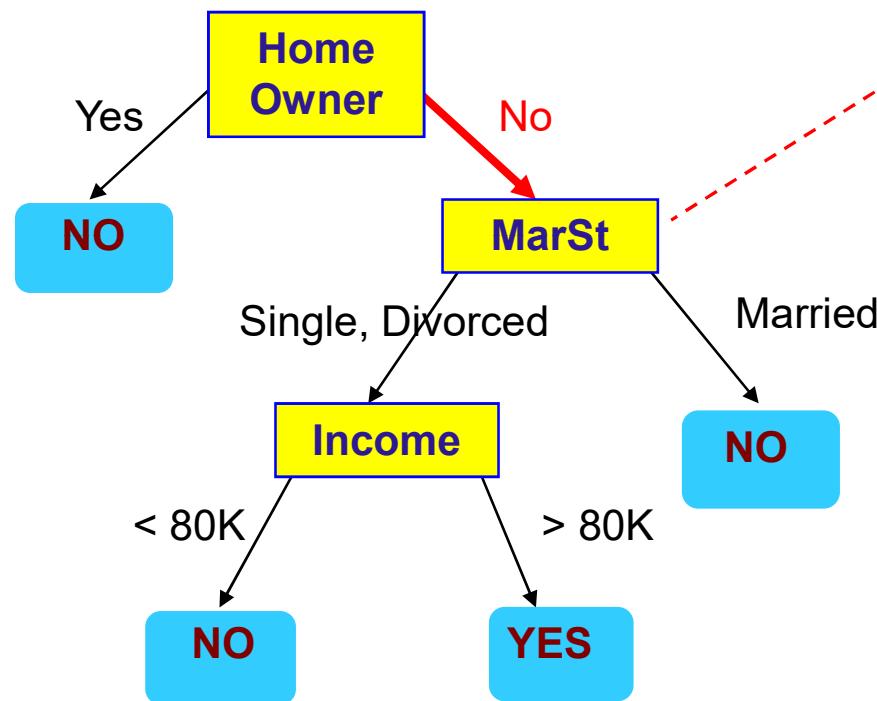
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

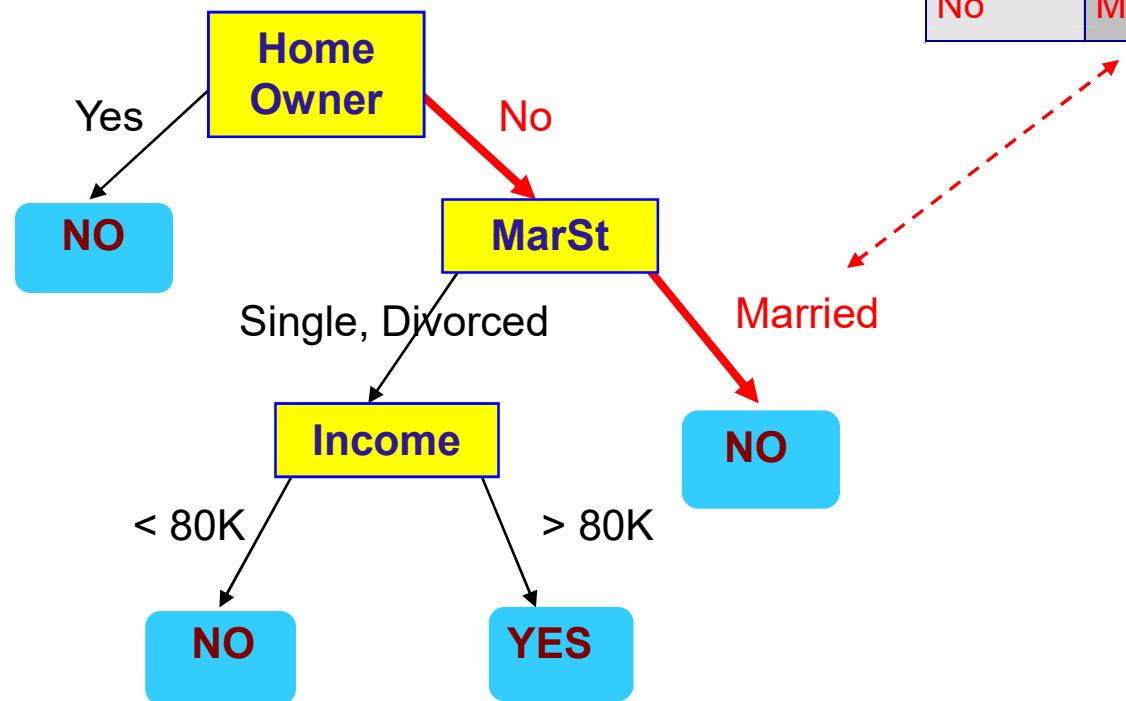
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

Test Data

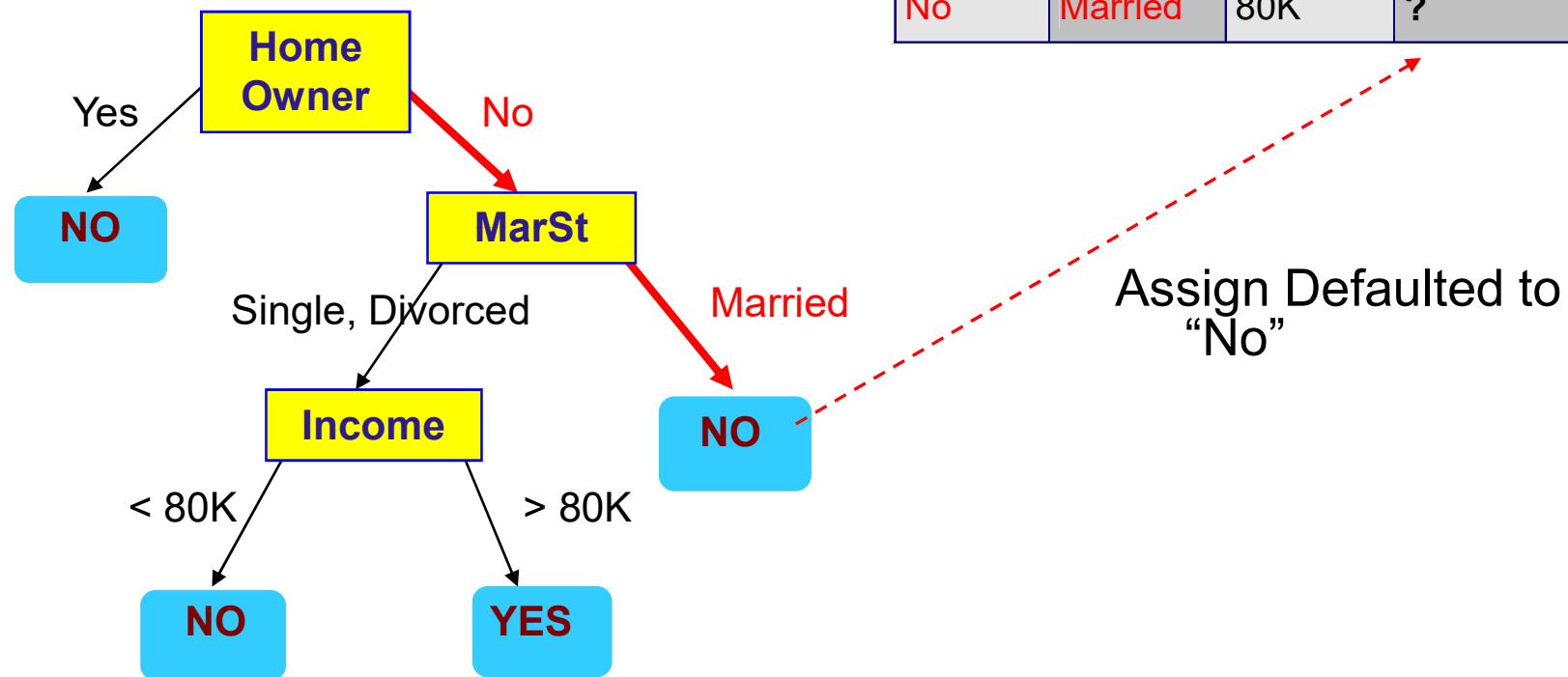
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

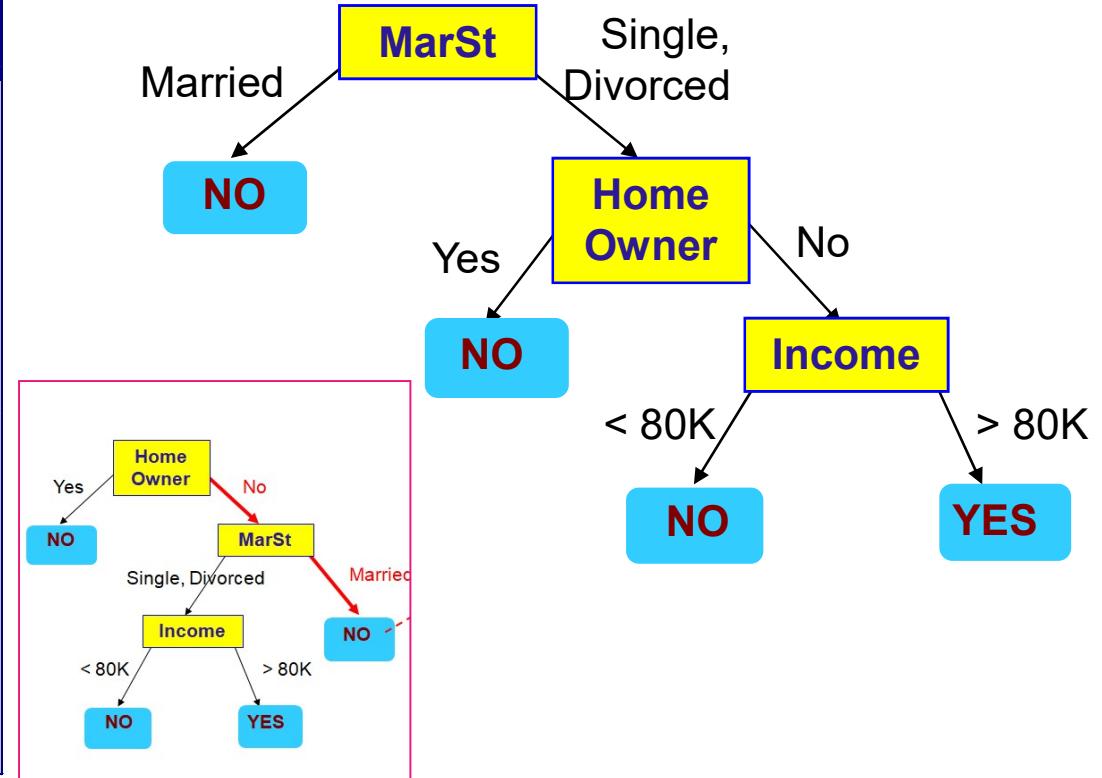
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Another Example of Decision Tree

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



There could be more than one tree that fits the same data!

درختی که میخواهد ایجاد بشه یا مدلی که میخواهد ایجاد بشه لزوماً یکتا نیست --> توی خیلی از مسائل Classifier همین است ینی حتی ترتیب نمونه ها هم چجوری باشه ممکنه یک درخت دیگه ای به ما بده و لزوماً به یک درخت نمی‌رسیم و یکسری شرایط خاص وجود داره که به یک درخت بررسیم

Decision Tree Induction

- Many Algorithms:
 - Hunt's Algorithm (one of the earliest)
 - CART
 - ID3, C4.5
 - SLIQ, SPRINT

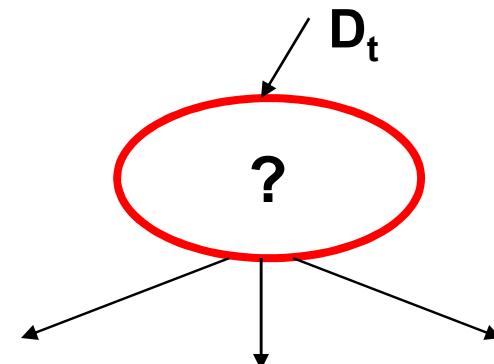
-
این درخت چطوری ایجاد میشه؟

الگوریتم های مختلفی رو او مدن اینجا توسعه دادن که این درخت رو برای ما پیدا بکنه به صورت خودکار

General Structure of 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 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.

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



- ساختاری که الگوریتم هانت دارد:

فرض میکنیم یک مجموعه دیتایی رو به ما دادن و به ما میگن یک ویژگی رو انتخاب بکن برای تصمیم گیری

حالا می خوایم مشخص بکنیم برای این ویژگی، شاخه هاش چی باشه و داخلش چی باشه --> دو تا مسئله برآش پیش میاد:

یا اون دیتایی که ما کامل دادیم برای این ویژگی همشون یک برچسب دارن و اگر همشون یک برچسب داشتن این ویژگی رو ینی نودی که الان باهاش سر و کار داریم رو تبدیل بکن به یک نود برگ و مقدارش رو بذار اون مقداری که متداول هست ینی مقداری که همه باهاش برابرند و اگر برچسب ها یکی نبودن ما باید یک ویژگی رو انتخاب بکنیم و بذاریم داخل این نود به عنوان یک نود داخلی که دو تا مسئله اینجا پیش میاد: یکی این که ما کدام یکی از این ویژگی ها رو بذاریم این داخل و مسئله دوم این که حالا هر کدام از این ویژگی ها رو که گذاشتیم چطوری این ها رو برنج بندی بکنیم ینی تقسیم شون بکنیم ینی کدام را بذاریم یک سمت و کدام را بذاریم یک سمت دیگه

Hunt's Algorithm

Defaulted = No

(7,3)

(a)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

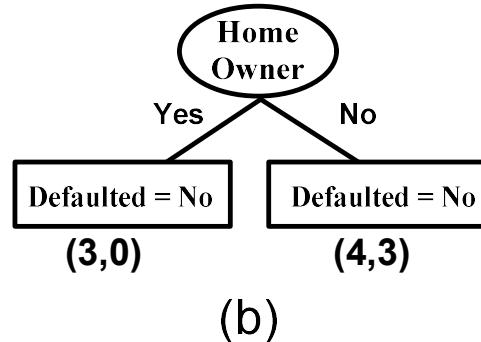
فرض کنیم الگوریتم هانت رو گذاشتیم و کل این دیتا رو بهش دادیم و در حالت پیش فرض اینو میداره no بینی از این 10 تا نمونه 7 تا نمونه برچسبش no بودن و 3 تا نمونه برچسبش yes و به صورت پیش فرض ما می تونیم بگیم که این برچسبش no است (a)

Hunt's Algorithm

Defaulted = No

(7,3)

(a)



(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

: b

حالا میخوایم دقیق خودمون رو توسعه بدیم و بهتر بگنیم --> میاد یک نود رو انتخاب میکنه مثلا home owner رو انتخاب کرد و این home owner دو تا حالت داره: یا فرد صاحب خونه هست یا نیست وقتی home owner رو به عنوان ویژگی انتخاب بگنیم بر حسب این که این home owner سمپل ها یس باشه یا نو باشه یکسری سمپل ها می افتن سمت راست و یکسری سمپل ها می افتن سمت چپ

حالا می رسیم به نودهای برگ --> صفحه بعدی...

Hunt's Algorithm

Defaulted = No

(7,3)

(a)

Home
Owner

Yes

No

Defaulted = No

Defaulted = No

+ (3,0)

(4,3) **

(b)

Home
Owner

Yes

No

Defaulted = No

(3,0) Single,
Divorced

Marital
Status

Married

Defaulted = Yes

(1,3)

(3,0) +

(c)

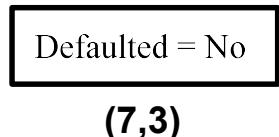
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

اساسمون برای برگ کردن یک نود چی بود؟ اگر همه نمونه هایی که توانی اون نود وجود دارن همشون یس یا نو باشن ینی همشون یک برچسب داشتن ما میتوانیم اونو برگش بکنیم الان این اتفاق واسه قسمت + افتاده

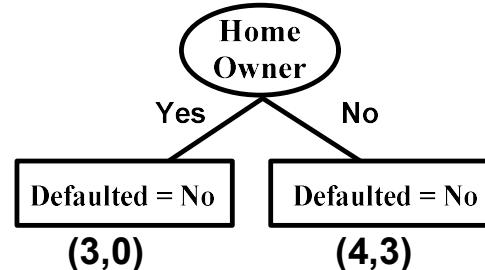
حالا باز لازمه که قسمت ** رو توسعه بدیم --> حالا اینجا باز ما باید یک ویژگی رو انتخاب بکنیم مثلا الان ویژگی marital status رو انتخاب میکنیم که خود این هم دو قسمت میشه: اونایی که ازدواج کردن و اونایی که نکردن و اینجا هم توانی بخش C قسمت + باز نویمون برگ میشه و اون یکی رو توسعه میدیم --> صفحه بعدی...

Hunt's Algorithm

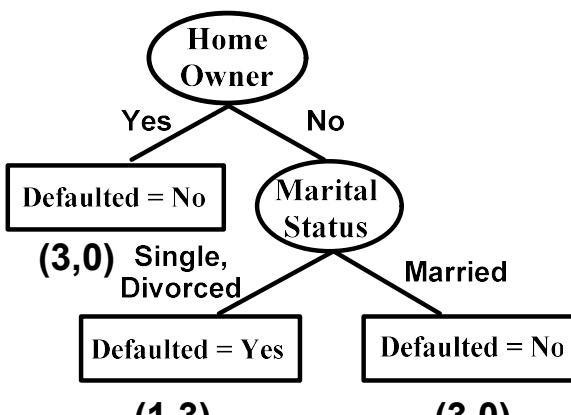
برچسبی که روی این دیتا متدائل هست no است پس به صورت پیش فرض اونو no در نظر میگیره



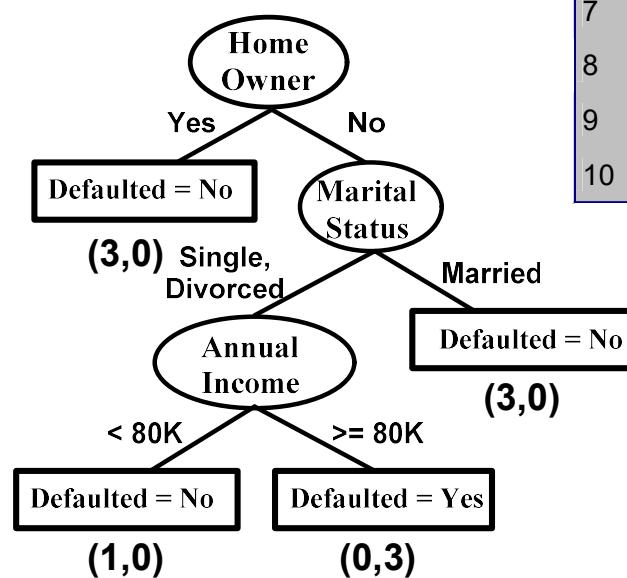
(a)



(b)



(c)



(d)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

d : مثل صفحه های قبلی اینم میشه

دوتا مسئله اینجا پیش میاد:

- 1- تا کجا باید این عملیات رو ادامه بدیم؟
- 2- بعضی از این ویژگی ها رو که میخوایم انتخاب بکنیم حالتشون باینری نیست و چندتایی است
مثل وضعیت تاہل که سه حالته است

توی برگ ها برچسب رو قرار می دیم

Design Issues of Decision Tree Induction

- How should training records be split?
(splitting criterion)
- How should the splitting procedure stop?
(stopping criterion)

سوال 2: عملیات هی متغیر اضافه کردن و تا تموم شدن رو تا کجا باید ادامه بدیم؟ (معیار توقف الگوریتم)

3 تا معیار داره که میگه به هر کدام از این 3 تا رسیدی می تونی متوقف بشی --> ادامش صفحه بعدی...

سوال 1: خود این ویژگی ها رو از کجا انتخاب بکنیم؟ ینی بر چه اساسی این ویژگی ها رو انتخاب بکنیم که درخت بهتر بشه و زودتر مسئله حل بشه؟ دسته بندی هاشون بر چه اساسی انتخاب بشه؟

نکته: درخت بهتر درختی است که تا اونجایی که میشه کوچیک باشه

Design Issues of Decision Tree Induction

- How should the splitting procedure stop?
(stopping criterion)
 - Stop splitting if **all the records belong to the same class** or have identical attribute values
 - There are **no remaining attributes** for further partitioning
 - There are **no samples left** – majority voting on the parent's samples is employed.

- 1- وقتی که همه رکوردها توی یک کلاس هستن ینی حالتی که توی مثال قبلی داشتیم ینی به جایی رسیدیم که توی اون شاخه همه رکوردها یک برچسب داشتن مثلا همشون برچسبشون یس شده بود یا نو شده بود

2- معیار دوم اینه که اصلا هیچ ویژگی یا attributes نمونه ینی همه ویژگی ها رو انتخاب کردیم و ویژگی دیگه ای نیست که بخوایم اضافه بکنیم

3- هیچ سمپل دیگه ای نمونه اینجا --> برچسبی که برای این قرار میدیم برچسب متداول است

Design Issues of Decision Tree Induction

- How should training records be split?
(splitting criterion)
 - Method for expressing test condition
 - ◆ depending on attribute types
 - Measure for evaluating the goodness of a test condition

-
معيار تقسيم:

Methods for Expressing Test Conditions

- Depends on attribute types
 - Binary
 - Nominal
 - Ordinal
 - Continuous

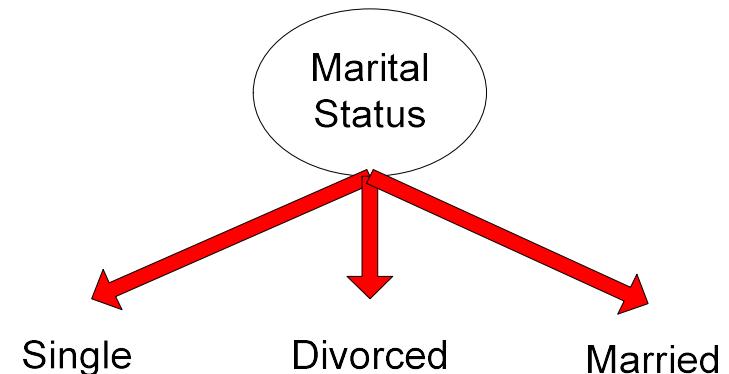
-
ویژگی هایی که توی دیتا داریم شامل این 4 تا است

وقتایی که با Nominal سر و کار داریم --> مثل وضعیت تاہل که توی این می تونیم چند حالت داشته باشیم مثلا هر کدوم رو بذاریم توی یک شاخه یا دوتا رو بذاریم توی یک شاخه و یکی دیگه رو بذاریم توی یک شاخه دیگه --> ما باید بین این چندتا تصمیم بگیریم که کدوم برای مدل ما بهتر است

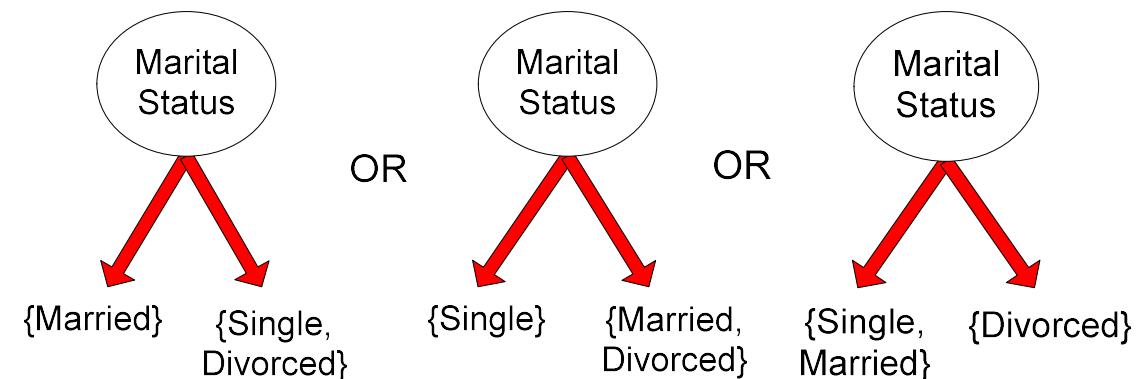
برای متغیر Continuous مشکل خیلی سخت میشه مثل سطح درآمد که اعدادش پیوسته هست-->
اینکه ما تشخیص بدیم کجا رو تقسیم بکنیم مثلا سطح درآمد بزرگتر از 80 بشه یک شاخه و کمتر از 80 بشه یک شاخه دیگه یا اینکه بیایم بازه بازه بکنیم

Test Condition for Nominal Attributes

- Multi-way split:
 - Use as many partitions as distinct values.

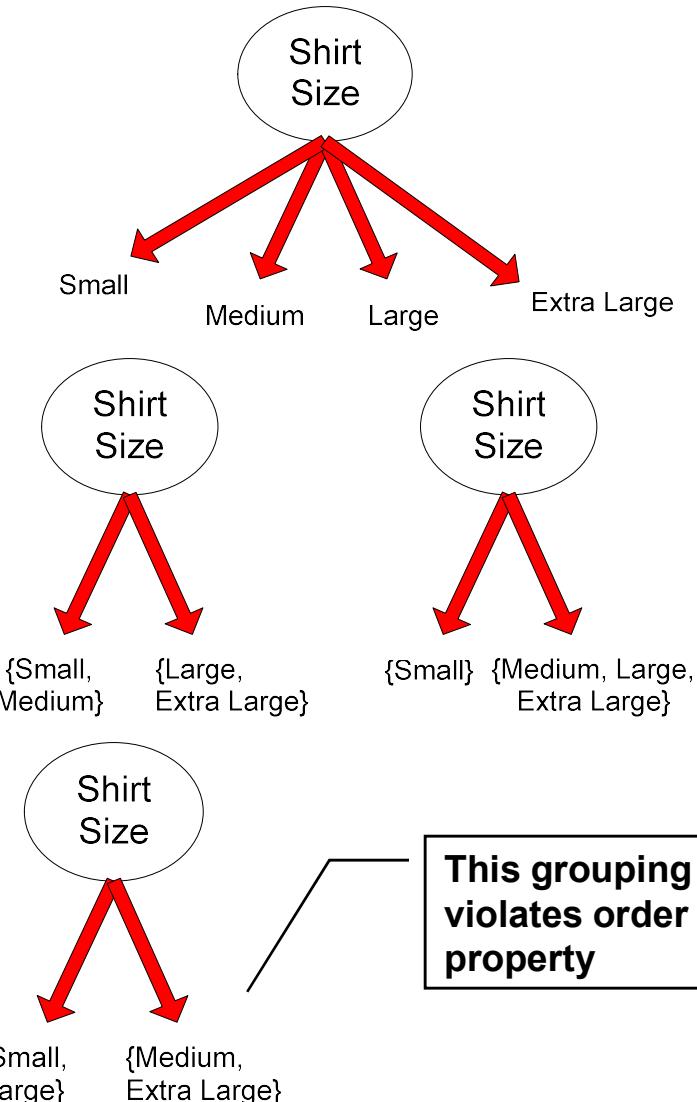


- Binary split:
 - Divides values into two subsets

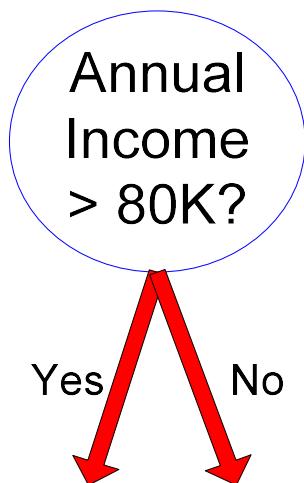


Test Condition for Ordinal Attributes

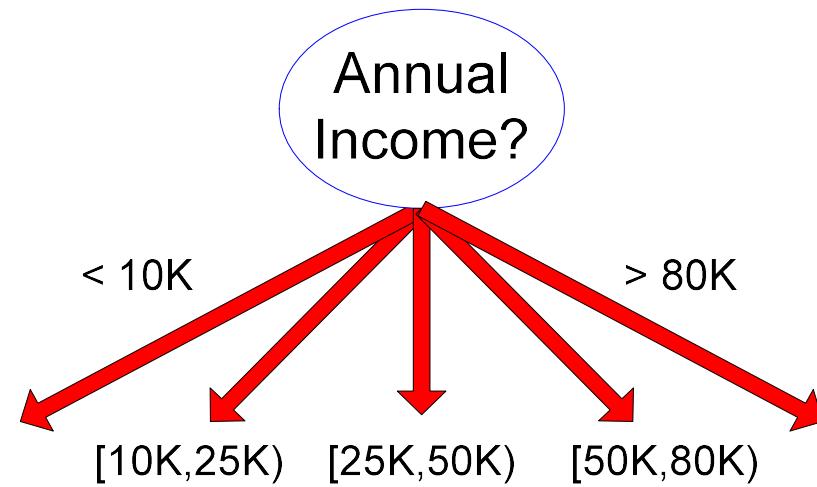
- Multi-way split:
 - Use as many partitions as distinct values
- Binary split:
 - Divides values into two subsets
 - Preserve order property among attribute values



Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

-

برای متغیرهای پیوسته میایم اونارو به چند دسته تقسیم میکنیم

Splitting Based on Continuous Attributes

- Different ways of handling
 - **Discretization** to form an ordinal categorical attribute

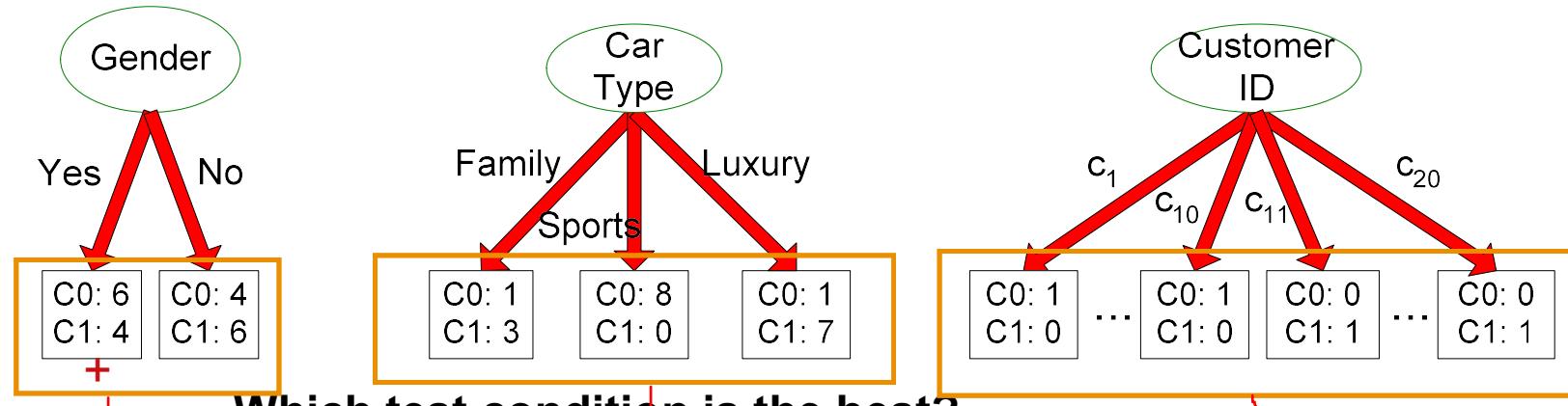
Ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

 - ◆ Static – discretize once at the beginning
 - ◆ Dynamic – repeat at each node
 - **Binary Decision:** $(A < v)$ or $(A \geq v)$
 - ◆ consider all possible splits and finds the best cut
 - ◆ can be more compute intensive

How to determine the Best Split

**Before Splitting: 10 records of class 0,
10 records of class 1**

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1



سه تا ویژگی داریم و یک همچین دیتاستی داریم:
حالا اینجا میخوایم یک ویژگی رو انتخاب کنیم به عنوان نود ریشه که اینجا سه تا ویژگی رو گفته
که باید یکی از این سه تا باشه، کدام یکی؟

نکته: اگر قرار بود + برگ بشه برچسبش باید اونی بشه که متداول است یعنی میشه C0 برای بقیه هم
به همین صورت می تونیم بگیم

برچه مبنایی بیایم بهترین ریشه رو انتخاب بکنیم؟ یکنواختی
یک ویژگی رو انتخاب می کنیم که اگر اون ویژگی رو مینا بذاریم توی هر شاخه ای بیشتر همه داده
ها بیو何ん توی یک کلاس --> یعنی شاخه ها یکنواتر بشن و یک دست تر بشن --> یعنی ناخالصیش
کمتر باشه یعنی دنبال ویژگی هایی هستیم که ما رو به فضایی بیره که ناخالصی کمتری داره
ناخالصی یعنی دوتا کلاس رو باهم نبینیم --> ترجیحا یک کلاس توی یک شاخه باشه که کار ما
راحت تر باشه
پس دنبال یک معیاری هستیم که ناخالصی رو برآمون بسنجه یعنی عدم قطعیت رو برآمون بسنجه

Measures of Node Impurity

- Gini Index

ماکریم مقدار gini وقتی که بیشترین ناخالصی وجود دارد

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Entropy

$$Entropy = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

- Misclassification error

$$Classification\ error = 1 - \max[p_i(t)]$$

3 تا معیار هست که میاد ناخالصی نودها رو برآمون اندازه گیری میکنه:

How to determine the Best Split

- Greedy approach:
 - Nodes with **purer** class distribution are preferred
- Need a measure of node impurity:

C0: 5
C1: 5

High degree of impurity

در این بیشترین ناخالصی وجود داره

C0: 9
C1: 1

Low degree of impurity

در این کمترین ناخالصی وجود داره

مثلا روشنی که برای ساخت درخت داریم به این صورت است که مانگاه بکنیم براساس هر کدام از این نودها مثلا چندتا ویژگی داریم و یکیشونم خواهیم انتخاب بکنیم و ببینیم کدام یکی از این ویژگی ها ناخالصی کمتری بهمون میده و اونو انتخاب بکنیم و دوباره اونو مبنا بذاریم و اون ویژگی رو نگاه بکنیم ببینیم نودهای برگش چطوری میشه و همین عملیات را برای نودهای برگش انجام بدیم --> این رویه که ما برای ساخت درخت داریم به صورت حریصانه است یعنی قرار نیست ما برگردیم و درستش بکنیم و هی به سمتی می ریم که ویژگی هایی که انتخاب میشن ناخالصی درخت ما رو هرچی به برگ نزدیک می شیم کمتر بکنه

Finding the Best Split

1. Compute impurity measure (P) before splitting
2. Compute impurity measure (M) after splitting
 - Compute impurity measure of each child node
 - M is the weighted impurity of child nodes
3. Choose the attribute test condition that produces the highest gain

$$\text{Gain} = P - M$$

or equivalently, lowest impurity measure after splitting (M)

- فرایندی که برای ساخت درخت داریم به صورت زیر است:

همیشه یه میزان ناخالصی رو قبل از اضافه کردن نود حساب میکنیم $P =$

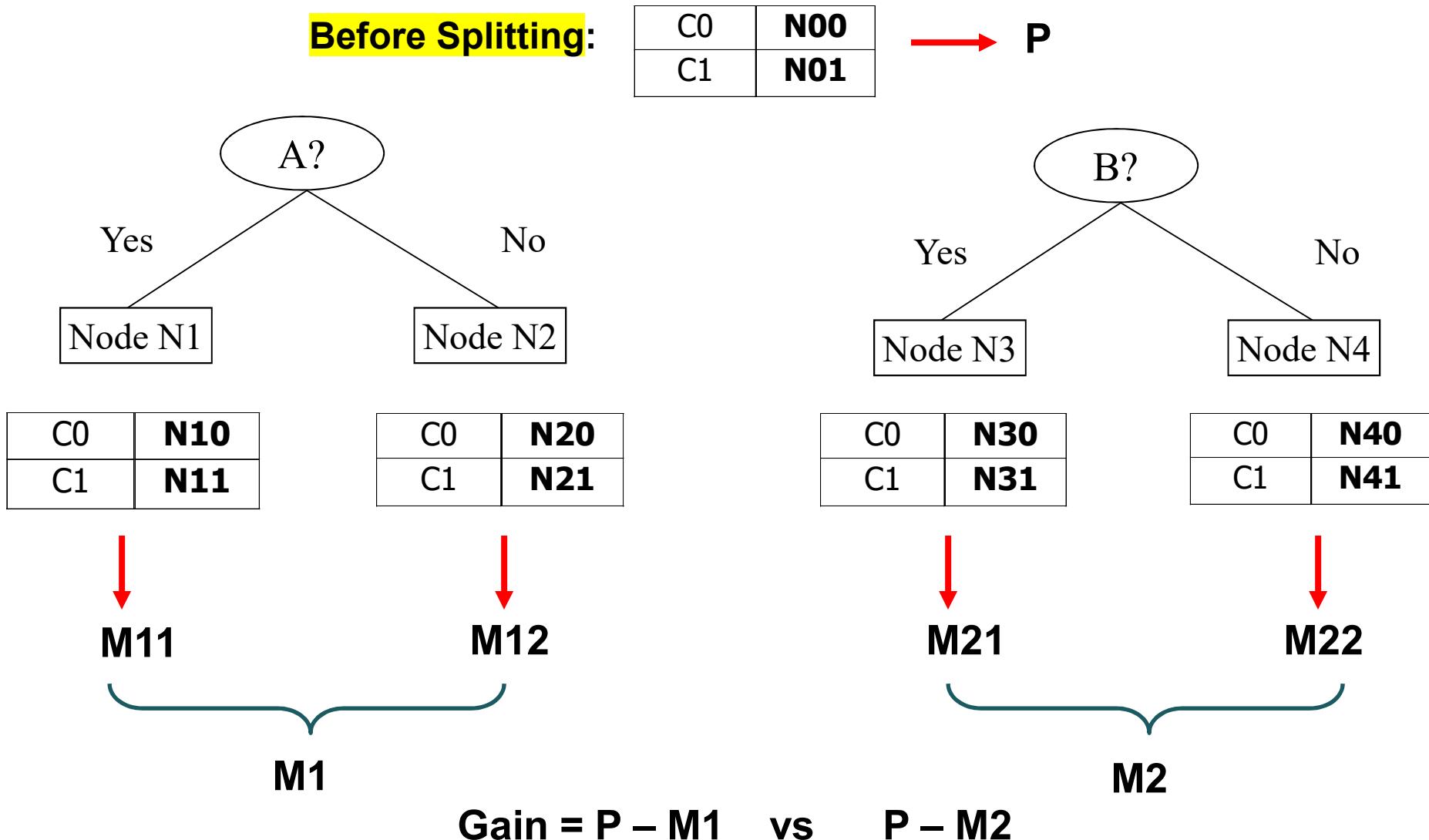
بعد که نود رو اضافه کردیم بازم ناخالصی رو حساب میکنیم $M =$

اختلاف این ها بهمن یک Gain میده ینی بهمن میگه چقدر رفتیم به سمت ناخالصی کمتر و دنبال

نودی هستیم که ناخالصی ما رو هرچی میشه کمتر بکنه که به این میگیم Gain پس به دنبال

نودی هستیم که Gain بیشتری داره

Finding the Best Split



Measure of Impurity: GINI

- Gini Index for a given node t

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
- Minimum of 0 when all records belong to one class, implying the most beneficial situation for classification

© 2021, Sharad Mehrotra. All rights reserved.

$P_i(t)$ احتمال یا فرکانس هر کدوم از کلاس ها است مثلا اگر دو تا کلاس داریم ینی P_1 ، P_2 بیا احتمال هر کدوم از این ها رو حساب بکن و توان دو اینا رو بگیر و بعد جمعشون بکن و نتیجه رو از یک کم بکن

مینیمم Gini زمانی است که: اگر به یک نودی بربخوریم که همه داده های مرتبط با اون نود یک کلاس باشن و از کلاس دوم هم هیچ رکوردی رو نبینیم در این حالت Gini ما صفر است--> این حداقل مقداری است که Gini میده

ماکزیمم Gini: وقتی که کلاسها همسوون فرکانشون با هم یک باشه ینی اگر 30 تا کلاس داریم 1/30 باشه

Measure of Impurity: GINI

- Gini Index for a given node t :

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

- For 2-class problem ($p, 1 - p$):
 - ◆ $\text{GINI} = 1 - p^2 - (1 - p)^2 = 2p(1-p)$

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	

Computing Gini Index of a Single Node

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Gini} = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Gini} = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

محاسبه شاخص جینی یک گره

Computing Gini Index for a Collection of Nodes

- When a node p is split into k partitions (children)

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

where, n_i = number of records at child i ,
 n = number of records at parent node p .

وقتی که یک ویژگی رو انتخاب میکنیم این ویژگی میاد چند شاخه تولید میکنه و هر شاخه ای هم یک Gini داره و بعد باید بیایم این Gini ها رو با هم ترکیب بکنیم و برای این که بیایم این Gini ها رو با هم ترکیب بکنیم از یک معیاری به نام Gini split استفاده میکنیم این معیار میاد Gini شاخه های مختلف رو با هم ترکیب میکنه فرمولش میاد به صورت وزن دار Gini ها رو با هم ترکیب میکنه --> ینی میگه توی هر شاخه ای چندتا کلا رکورد است و بعد بیا Gini رو با همون وزن اضافه بکن

Binary Attributes: Computing GINI Index

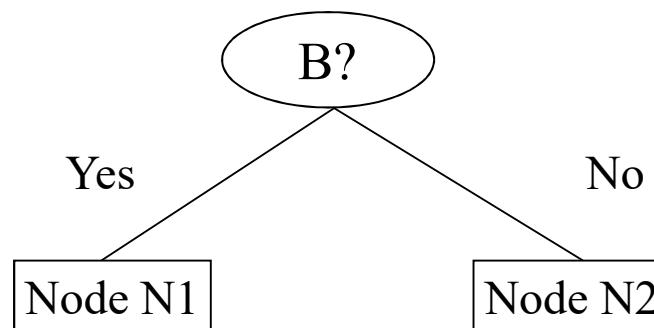
- Splits into two partitions (child nodes)
- Effect of Weighing partitions:
 - Larger and purer partitions are sought

Gini(N1)

$$\begin{aligned} &= 1 - (5/6)^2 - (1/6)^2 \\ &= 0.278 \end{aligned}$$

Gini(N2)

$$\begin{aligned} &= 1 - (2/6)^2 - (4/6)^2 \\ &= 0.444 \end{aligned}$$



	N1	N2
C1	5	2
C2	1	4
Gini=0.361		

	Parent
C1	7
C2	5
Gini = 0.486	

Weighted Gini of N1 N2

$$\begin{aligned} &= 6/12 * 0.278 + \\ &\quad 6/12 * 0.444 \\ &= 0.361 \end{aligned}$$

$$\text{Gain} = 0.486 - 0.361 = 0.125$$

Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

می تونیم هر کدام از اینا رو یک شاخه ای بکنیم که میشه سمت چپی یا اینکه دو تا شاخه ای بکنیم که میشه سمت راستی:

Multi-way split

CarType			
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

Two-way split

(find best partition of values)

CarType		
	{Sports, Luxury}	{Family}
C1	9	1
C2	7	3
Gini	0.468	

CarType		
	{Sports}	{Family, Luxury}
C1	8	2
C2	0	10
Gini	0.167	

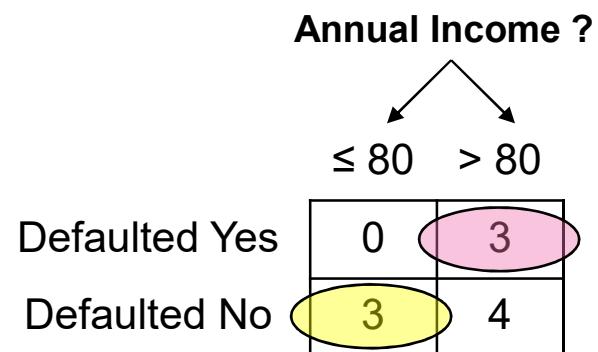
Which of these is the best?

هر کدام از این ها یک Gain میده --> در نهایت اون Gain رو انتخاب میکنیم که اش بیشتر است
ولی Gini باید کمتر باشه

Continuous Attributes: Computing Gini Index

- Use Binary Decisions based on one 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, $A \leq v$ and $A > v$
- 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.

ID	Home Owner	Marital Status	Annual Income	Defaulted
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



-
ویژگی های پیوسته:

می تونیم گزینه هایی که برای تقسیم بندی کردن وجود داره رو در نظر بگیریم مثلا یکیش 125k است پس میشه قبل این و بعد این - گزینه بعدی 100k است - گزینه بعدی 70k و به همین صورت می تونیم گزینه داشته باشیم تا پایین --> پس می تونیم تک تک این ها رو برای مباذازیم و بباییم Gain را حساب بکنیم و هر کدام Gain بهتری داد بباییم اونو انتخاب بکنیم

Continuous Attributes: Computing Gini Index...

- 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

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No	No
Annual Income											
Sorted Values	→	60	70	75	85	90	95	100	120	125	220

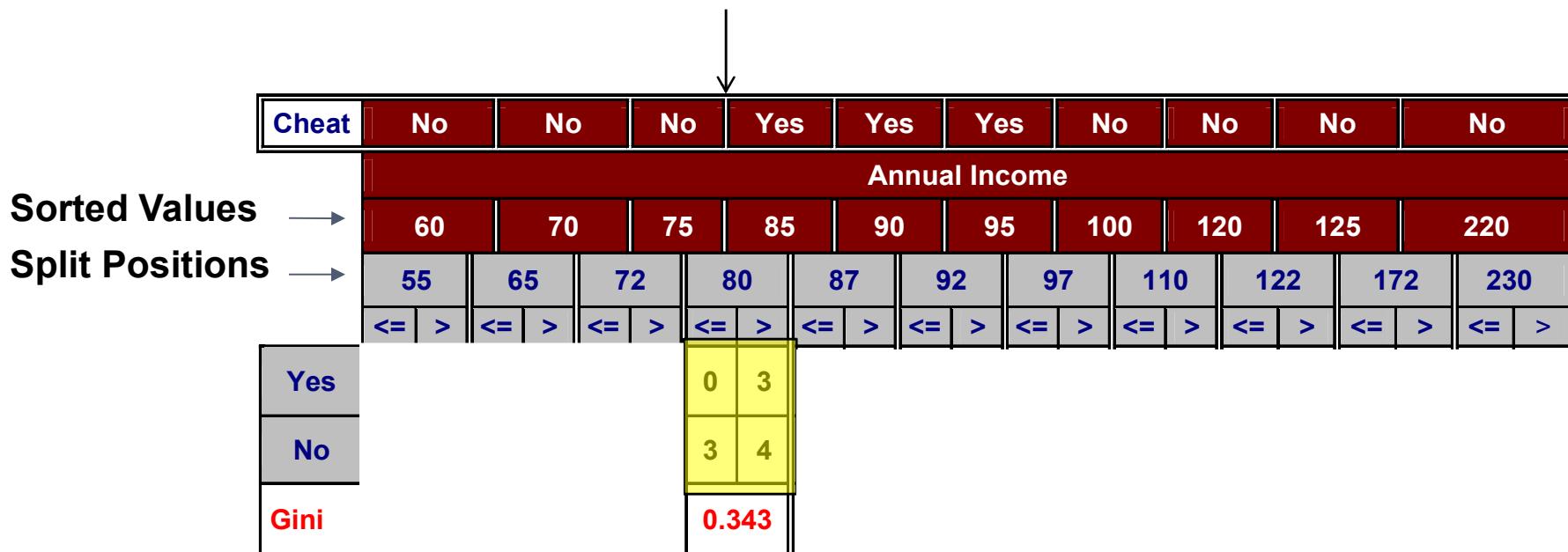
Continuous Attributes: Computing Gini Index...

- 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

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Annual Income										
Sorted Values	60	70	75	85	90	95	100	120	125	220
Split Positions	55	65	72	80	87	92	97	110	122	172
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >

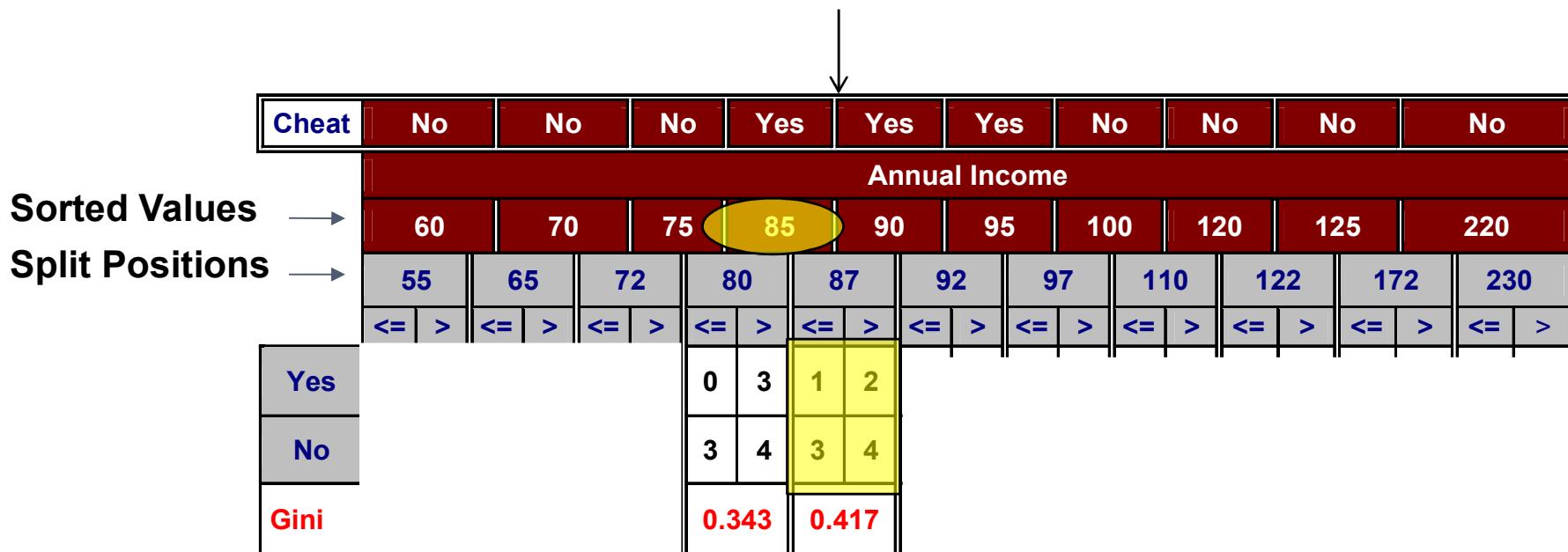
Continuous Attributes: Computing Gini Index...

- 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



Continuous Attributes: Computing Gini Index...

- 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



Continuous Attributes: Computing Gini Index...

- 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

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No	No
Annual Income											
Sorted Values →	60	70	75	85	90	95	100	120	125	172	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

Measure of Impurity: Entropy

Entropy at a given node t

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- ◆ Maximum of $\log_2 c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
- ◆ Minimum of 0 when all records belong to one class, implying most beneficial situation for classification
- Entropy based computations are quite similar to the GINI index computations

Entropy: میزان بی نظمی رو اندازه گیری می کرد

اینجا برای Entropy هم مشابه gini است یعنی یک Entropy قبل از تقسیم بندی داریم و یک هم بعد از تقسیم بندی که اختلاف اینها gain رو میده Entropy

Computing Entropy of a Single Node

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Entropy} = -(1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$$

C1	2
C2	4

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Entropy} = -(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

محاسبه آنتروپی یک گره منفرد

Computing Information Gain After Splitting

Information Gain:

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

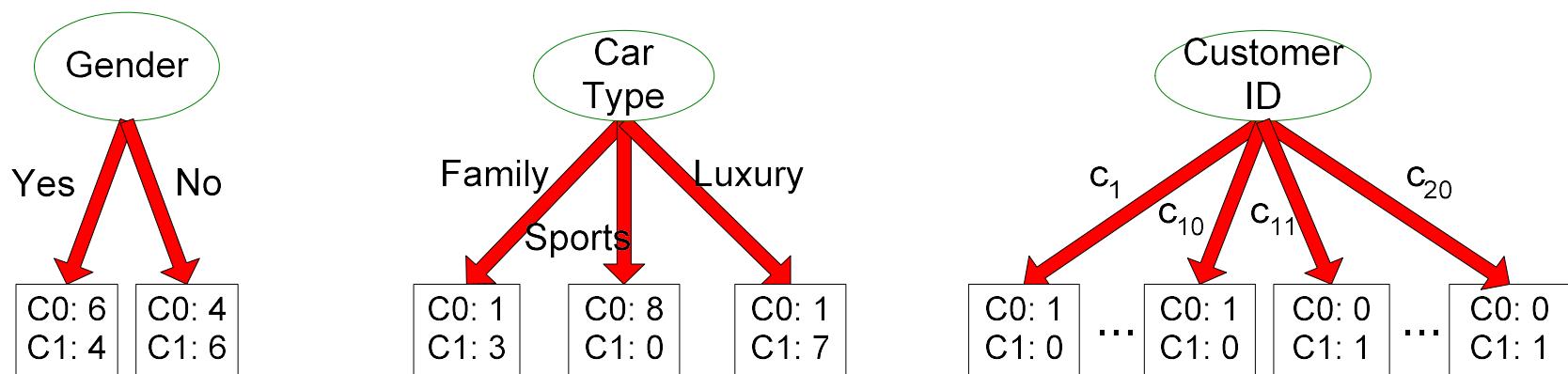
Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

- Choose the split that achieves most reduction (maximizes GAIN)
- Used in the ID3 and C4.5 decision tree algorithms
- Information gain is the mutual information between the class variable and the splitting variable

Problem with large number of partitions

- Node impurity measures tend to prefer splits that result in large number of partitions, each being small but pure



- Customer ID has highest information gain because entropy for all the children is zero

- وقتی با این ناخالصی کار بکنیم یک مشکلی پیش میاد اونم برای همین customer id است اگر میزان gain رو روی این ها مثل customer id حساب بکنیم ممکنه به یک شرایطی بررسیم که این customer id رو انتخاب بکنیم --> چون درخت ما رو به یه جایی می بره که شاخه هاش دیگه قطعیت دارن و هیچ ناخالصی توی هیچ کدام از شاخه هاش وجود نداره پس ترجیح می ره سمت این که این ویژگی انتخاب بشه

این یک مشکل است ولی مشکلی است که میشه حلش کرد --> می خوایم یه جوری جریمه بکنیم ینی اگر ویژگی رو انتخاب کردیم که خیلی شاخه داشت ببایم ارزش gain که اون میده رو کم بکنیم ینی جریمه اش بکنیم --> اینو اومدن چجوری حلش کردن؟ از همون مبنای Entropy کمک گرفتن و split info رو تعریف کردن

در حالت کلی customer id غلط است چون هیچ اطلاعاتی تو ش نیست چون شماره دانشجویی هیچ ربطی به نمره نداره ینی کلا هیچ ربطی به برچسبش نداره --> ????

Gain Ratio

- Gain Ratio:

$$\text{Gain Ratio} = \frac{\text{Gain}_{\text{split}}}{\text{Split Info}} \quad \text{Split Info} = - \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

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

n_i is number of records in child node i

- Adjusts Information Gain by the entropy of the partitioning (*Split Info*).
 - ◆ Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5 algorithm
- Designed to overcome the disadvantage of Information Gain

داره همین کارو میکنه ینی میاد یه جوری تعداد زیاد شدن شاخه ها رو کمی می کنه و جریمه میکنه ینی وقتی تعداد شاخه ها زیاد میشه این split info خیلی عدد بزرگی میشه

در نهایت از Gain Ratio استفاده میکنیم

این split info توی الگوریتم C45 استفاده میشه

اونی که gain بیشتری داره انتخاب میشه ینی اونی میشه که gini کمتری داشته حالا توی هر شاخه ما 4/20 داریم و 8/20 و 8/20 و تهش به split info می رسم که split info این بزرگتر از همه شده

Gain Ratio

- Gain Ratio:

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info}$$

$$Split\ Info = -\sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Parent Node, p is split into k partitions (children)
 n_i is number of records in child node i

تهش کدوم انتخاب میشه ؟؟؟

CarType			
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52

CarType			
	{Sports, Luxury}	{Family}	
C1	9	1	
C2	7	3	
Gini	0.468		

SplitINFO = 0.72

CarType			
	{Sports}	{Family, Luxury}	
C1	8	2	
C2	0	10	
Gini	0.167		

SplitINFO = 0.97

Measure of Impurity: Classification Error

- Classification error at a node t

$$Error(t) = 1 - \max_i[p_i(t)]$$

- Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least interesting situation
- Minimum of 0 when all records belong to one class, implying the most interesting situation

:Classification Error

مثلاً توی این شاخه می گیم عمدہ برچسب چی هست--> وقتی میگیم عمدہ برچسب چی هست و اونو مبنا قرار میدیم پس اونایی که کمتر هستن یعنی خطأ حساب میشن یعنی داریم خطأ اضافه می کنیم به کار

مثال صفحه بعد...

Computing Error of a Single Node

$$\text{Error}(t) = 1 - \max_i[p_i(t)]$$

C1	0
C2	6

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Error} = 1 - \max(0, 1) = 1 - 1 = 0$$

C1	1
C2	5

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Error} = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

C1	2
C2	4

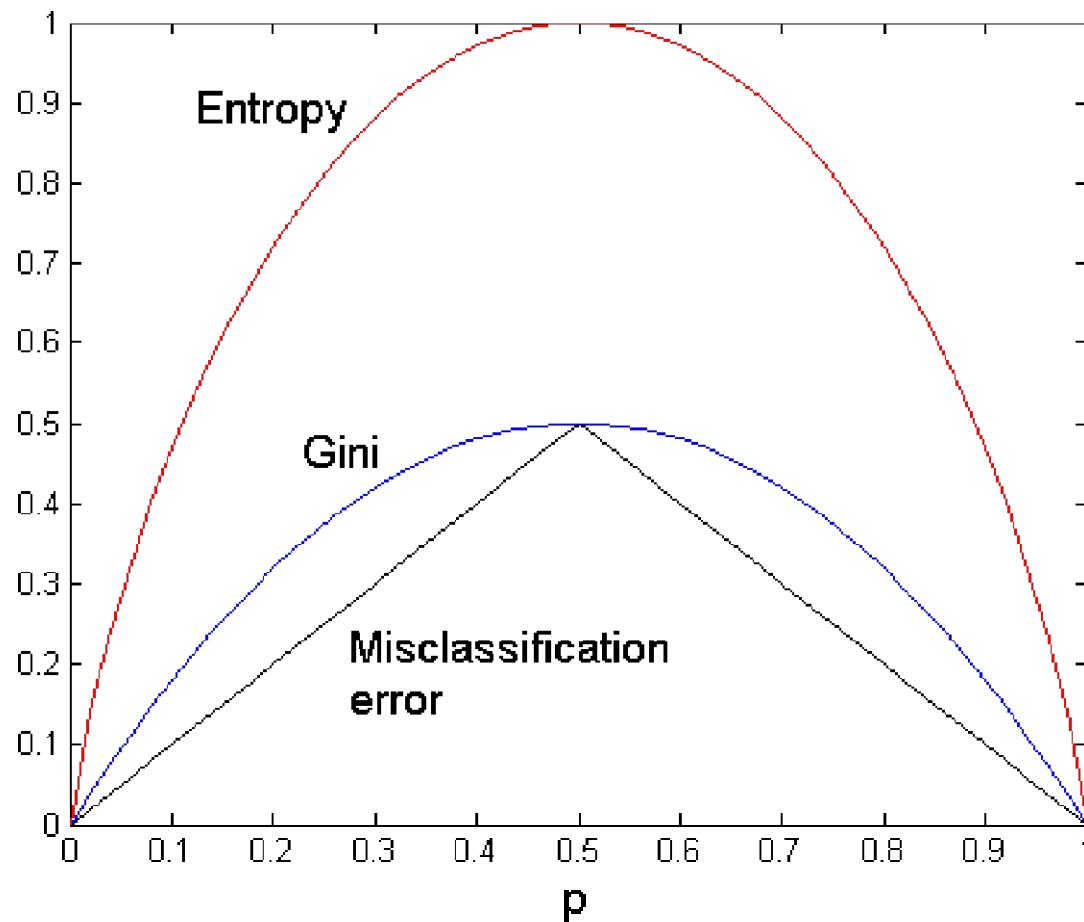
$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Error} = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

خطای محاسباتی یک گره واحد

Comparison among Impurity Measures

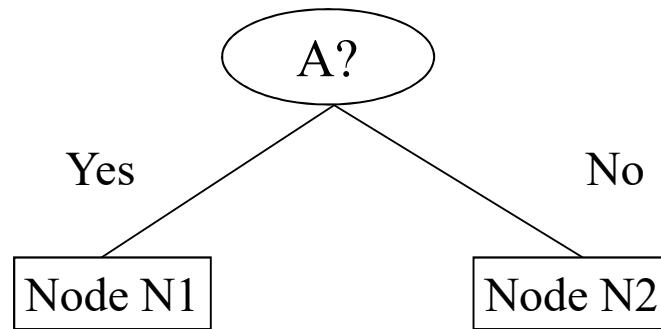
For a 2-class problem:



- مقایسه برای دو کلاس:

اگر احتمال یک کلاس P باشه اون یکی میشه یک منهای P
انتروپی توى 0.5 ماقزیم است --> این میشه اوج بى نظمی

Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

نود پدر ویژگی A

Gini(N1)

$$= 1 - (3/3)^2 - (0/3)^2 \\ = 0$$

Gini(N2)

$$= 1 - (4/7)^2 - (3/7)^2 \\ = 0.489$$

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

Gini(Children)

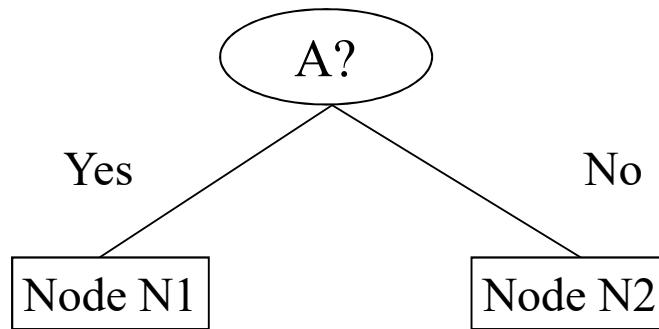
$$= 3/10 * 0 \\ + 7/10 * 0.489 \\ = 0.342$$

**Gini improves but
error remains the
same!!**

- اینجا $gini$ خوب شده ولی ارور تغییری نکرده چرا؟

قبل از اینکه ویژگی A را اضافه بکنیم ارور مون 3/10 است و بعد از اینکه ویژگی A را اضافه کردیم به امید اینکه این 3/10 را کمتر بکنیم --> بعد از اضافه کردن ویژگی A بازم خطای 3/10 است پس از نظر Misclassification Error این ویژگی اصلاً نباید اضافه بشه چون کمکی بهمون نمی‌کنه در حالی که اگر با شاخص $gini$ نگاه بکنیم از این نظر داره کار رو بهتر میکنه

Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

	N1	N2
C1	3	4
C2	1	2
Gini=0.416		

Misclassification error for all three cases = 0.3 !

Decision Tree Based Classification

- Advantages:

- Relatively inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Robust to noise (especially when methods to avoid overfitting are employed)
- Can easily handle redundant attributes
- Can easily handle irrelevant attributes (unless the attributes are **interacting**)

- Disadvantages: .

- Due to the greedy nature of splitting criterion, **interacting** attributes (that can distinguish between classes together but not individually) may be passed over in favor of other attributed that are less discriminating.
- Each decision boundary involves only a single attribute

مزایا و معایب درخت تصمیم:

مزایا:

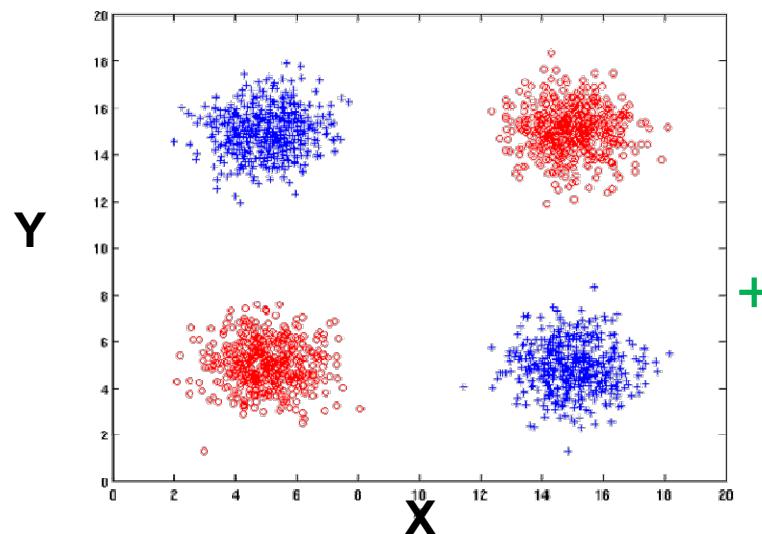
درخت تصمیم رو خیلی سریع می شه ایجاد کرد
اگر درخت کوچیک باشه قابل تفسیر است
نسبت به نویز مقاوم هستن

بعضی وقتا توی مسائل ویژگی هایی داریم که این ویژگی ها مثل هم هستن مثل اطلاعات دانشجوها رو گرفتیم مثل توی یک ستونی سنشون رو داریم و توی یک ستون دیگه تاریخ تولدشون رو داریم این سن و تاریخ تولد از لحاظ رفتاری مشابه هم هستن پس میگیم این دوتا ویژگی، ویژگی های وابسته هستن یعنی افزایش اون یکی همراه است پس اینجا میگیم با ویژگی های redundant سر و کار داریم یعنی ویژگی هایی که بهم ربط دارن --> این ویژگی ها وقتی بهم ربط پیدا بکنند رو گیج می کنند یعنی مدل اینجا ممکنه به اشتباه این دوتا ویژگی رو در نظر بگیره برای دسته بندی خودش و این خیلی مطلوب نیست --> اگر قرار دوتا ویژگی مثل هم رفتار بکنند ما ترجیح میدیم یکیشون رو داشته باشیم و اون یکی رو حذف بکنیم --> درخت های تصمیم از جمله روش هایی هستن که ما اگر ویژگی های مشابه هم بهشون بدیم اون ویژگی های مشابه رو شناسایی میکنند و یکیشون رو انتخاب می کنند و سراغ ویژگی دوم نمی روند:
(سرگرمی ادما و سن ادما مثل سرگرمی دهه 80 میشه بازی) --> مثالی که سر کلاس زد برای بهتر فهمیدن

وقت هایی که ویژگی ها هیچ ربطی بهم ندارند نسبت به اینا هم درخت تصمیم خوب عمل میکنند مثل سن با جنسیت مگر این که این ویژگی های ما حالت interacting یا برهم کنش داشته باشن معاایب:

همین interacting است یعنی وقتایی که ویژگی های ما حالت interacting هستن این درخت تصمیم خوب عمل نمیکند

Handling interactions



+ : 1000 instances

o : 1000 instances

Entropy (X) : 0.99

Entropy (Y) : 0.99

مثلا ریشه رو گرفتیم x و الان گفتیم برای 10 بیشتر و 10 کمتر ولی این مقدار 10 که انتخاب کردیم هیچ تاثیری نداشت برای ناخالصی کمتر

این مهمه!! چون جنس امتحان از این نوع دست است ینی تحلیل کردن میشه

الان چالشی که درخت تصمیم توی این حالت داره اینه که مثلا به ازای $x=6$ هم یکسری نمونه داریم که کلاسشون یک است و یکسری نمونه داریم که کلاسشون دو است

تصمیم گیری کمتر و بیشتر از 10 موجب خالص تر شدن تصمیم‌تون نمیشه--> اوج ناخالصی رو داره و اوج ناخالصی ینی بیشترین انتروپی

از بعد لا هم باز همین ماجرا است

درخت تصمیم فقط داره با یک بعد به مسئله نگاه می کنه ینی یک بعد درخت رو توسعه میده ینی اول یک ویژگی رو انتخاب میکنه و امیدواره که این ویژگی یکم خالص ترش بکنه بعد نتیجه هرچی شد نتیجه رو دوباره می ده برای ویژگی بعدی که می تونه

- کی میگیم دوتا ویژگی با هم interactions دارن؟ وقتایی که یک همچین فضایی + داریم: اینا یک مجموعه داده هستن که دوتا ویژگی ازشون جمع اوری شده یک ویژگی X است و یک ویژگی Y و دوتا کلاس هم داریم توی این داده هامون کلاس ابی و کلاس قرمز که از هر کدام 1000 تا نمونه داریم

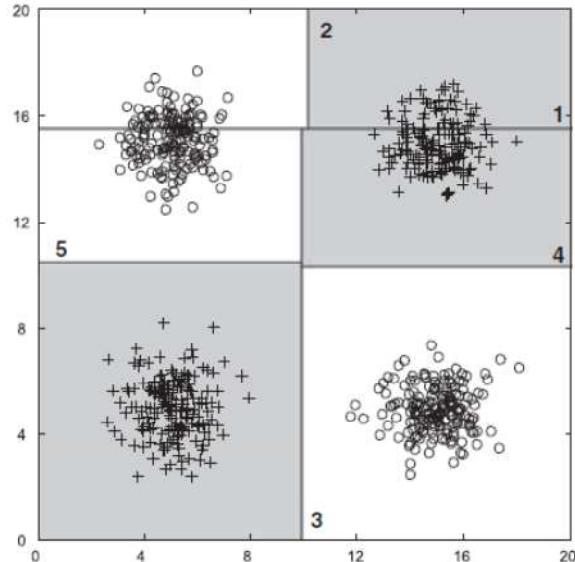
اگر درخت تصمیم بخواهد روی این داده ها اعمال بشه مثلًا متغیر X یا متغیر Y را بخواهد انتخاب بکنه در این حالت برای انتخاب کردن باید چی کار بکنه؟ مثلًا میاد انتروپی رو حساب میکنه میگه انتروپی X چدره و انتروپی Y چدره یکی از این ویژگی ها رو باید بذاریم ریشه --> حالا ویژگی های ما پیوسته هستن و درخت ما نمی تونه پیوسته کار بکنه و باید بشکنه این ویژگی رو

ادامه متن قرمز:

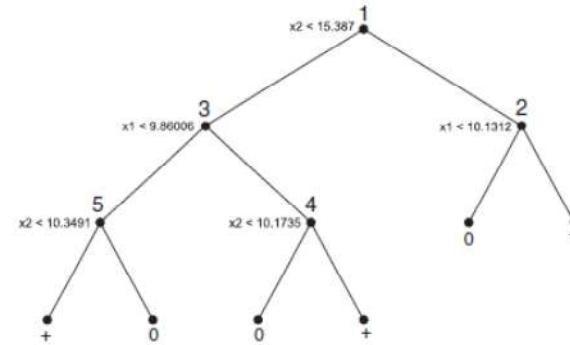
فرض میکنیم ویژگی X رو انتخاب کردیم بعد یک مقداری رو انتخاب میکنه مثلًا این مقدار 10 است که ما بعید می دوئیم این مقدار رو انتخاب بکنه چون اوج بی نظمی در 10 است --> مثلًا اینجا فرض میکنیم همین 10 رو انتخاب کرده که کمترش هم قرمز داریم و هم ابی و بیشترش هم باز به همین صورت است --> حالا توی ساخت درخت میگه کدوم ویژگی رو باید انتخاب بکنیم و با چه مقداری و به همین صورت می ره جلو حالا چرا این اتفاق افتاد و اسه درخت تصمیم ما؟ یعنی این داده ها چه ویژگی داشتن که این مسئله پیش آومد؟

دلیل مشکل اینه که ما برای اینکه بخوایم این داده ها رو جدا بکنیم باید همزمان دوتا متغیرشون رو ببینیم ولی درخت تصمیم این کارو نمیکنه چون یک بعد یک ره جلو --> درخت تصمیم میاد یکسری خط میکشه و فضا رو با یکسری خط های صاف جدا میکنه و خط های ما نمی تونه بریده بریده باشه یا کج باشه --> اگر ویژگی interactions داشته باشیم درخت تصمیم نسبت به اینا داستان برآش پیش میاد یه ذره

Handling interactions



(a) Decision boundary for tree with 6 leaf nodes.

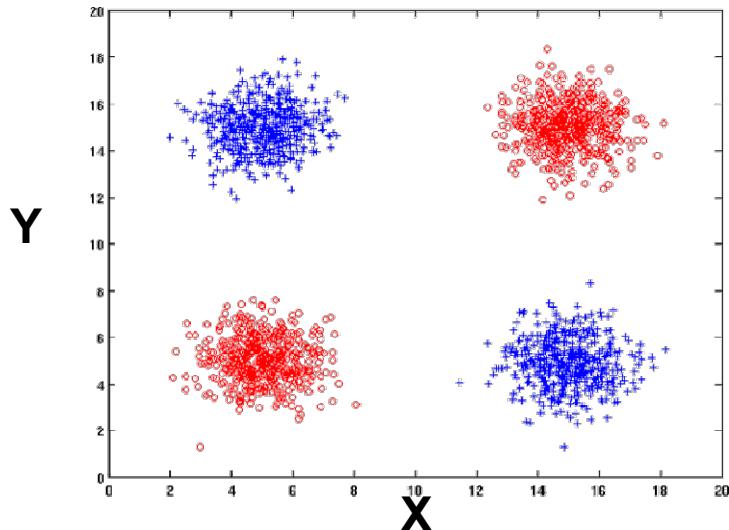


(b) Decision tree with 6 leaf nodes.

Figure 3.28. Decision tree with 6 leaf nodes using X and Y as attributes. Splits have been numbered from 1 to 5 in order of other occurrence in the tree.

مسئله ای که برای interactions ها پیش میاد توی درخت تصمیم این است که این درخته بزرگتر میشه

Handling interactions given irrelevant attributes



+ : 1000 instances

o : 1000 instances

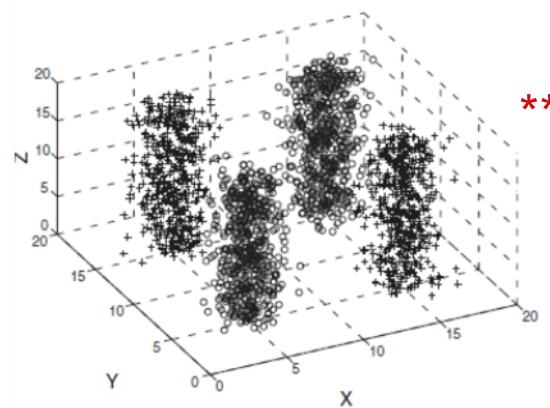
Adding Z as a noisy attribute generated from a uniform distribution

Entropy (X) : 0.99

Entropy (Y) : 0.99

Entropy (Z) : 0.98

Attribute Z will be chosen for splitting!



(a) Three-dimensional data with attributes X , Y , and Z .

-
توی **:

دوتا کلاس + و دایره داریم و سه تا ویژگی:

اینو میدیم به درخت تصمیم

به عنوان یک ناظر انسانی می دوینیم که بعد Z هیچ کمکی برای دسته بندی به این داده ها نمیکنند

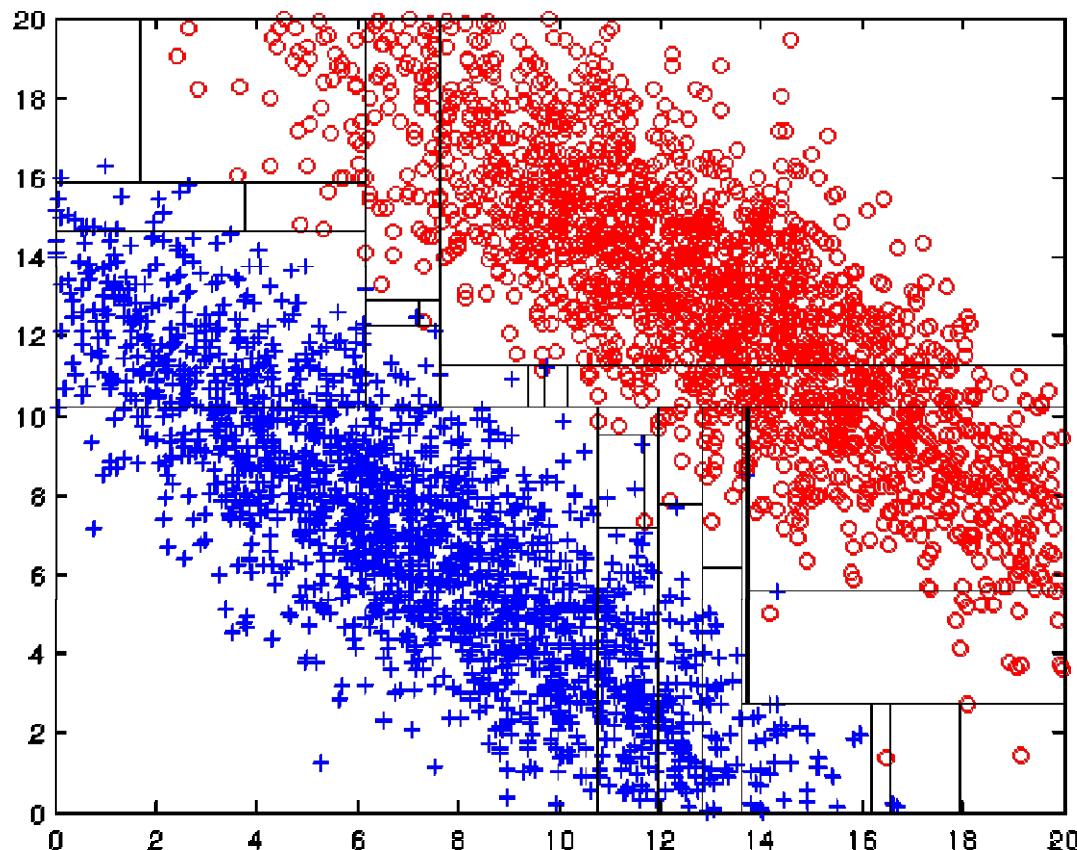
اینجا چون انتروپی Z کمتر است پس Z رو می ذاره مبنا به اشتباه پس ویژگی رو می ذاره مبنا که

ویژگی موثری نیست و بعدها می فهمه و سعی میکنند اشتباه رو جبران بکنند ولی درختش بزرگ شده

چون داره به سمتی حرکت می کنه که خطا رو کمینه بکنه

Limitations of single attribute-based decision boundaries

اینجا برای درخت تصمیم یه عالمه خط کشیده



Both **positive (+)** and **negative (o)** classes generated from skewed Gaussians with centers at (8,8) and (12,12) respectively.

محدودیت های مرز های تصمیم مبتنی بر ویژگی منفرد