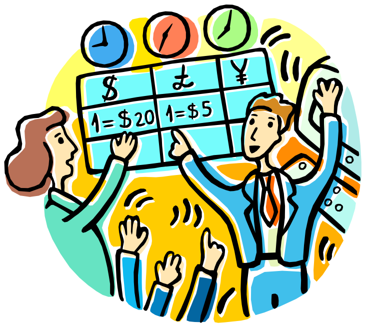
Lecture No. 4 – Decision Tree Learning I

- Classification and Prediction
- Overview of Decision Tree Learning
- Avoiding Overfitting

Classification vs. Prediction

- Classification
  - predicts categorical class labels (discrete or nominal)
  - Default predicted class: *majority voting*
  - Optional: class probability estimation

- Prediction / Regression
  - models continuous-valued functions, i.e., predicts unknown or missing values
  - Default prediction: *expected value*
  - Optional: confidence interval



Examples of typical classification / prediction tasks

- Typical classification tasks
  - Credit approval: approve / deny
  - Target marketing: will buy / will not buy
  - Medical diagnosis: Hepatitis B / Hepatitis C
  - Fraud detection: lawful transaction / fraudulent transaction
- Typical prediction / regression tasks
  - Weather forecast: predict tomorrow’s temperature
  - Stock trading: predict stock’s price tomorrow

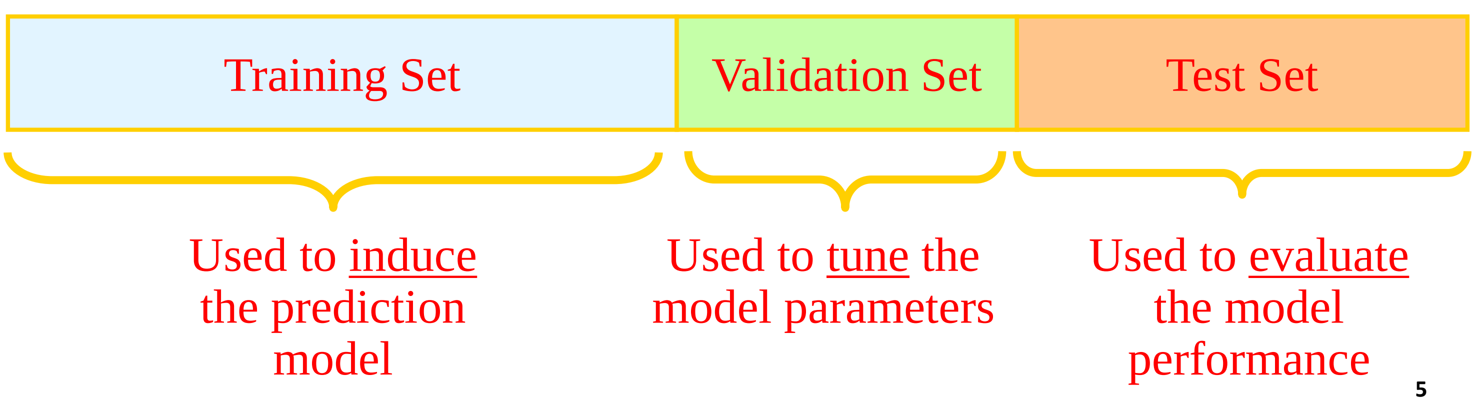
Classification—A Two-Step Process

- Model construction / induction
  - Record labeling
  - Building a *training set*
  - Inducing the model(s)
- Model usage
  - Comparing to the *default (majority) rule*
  - Measuring the accuracy rate over time

Training Set	Test Set	Score Set
Used to induce the classification model	Used to estimate the error rate of the induced model	Future objects to be classified by the model

Model Evaluation and Selection

- Evaluation metrics: How can we measure accuracy? Other metrics to consider?
- Use **test set** of class-labeled tuples instead of training set when assessing accuracy
  - The model should generalize beyond the training instances
- Use **validation set** to tune model parameters
  - Common splits: 50:20:30 or 40:20:40



- ***k*-Fold Cross Validation**

Process (1): Model Construction

Training
ta

Target (class) attribute

What is the accuracy of the majority rule?

What is the accuracy of this model on the training data?

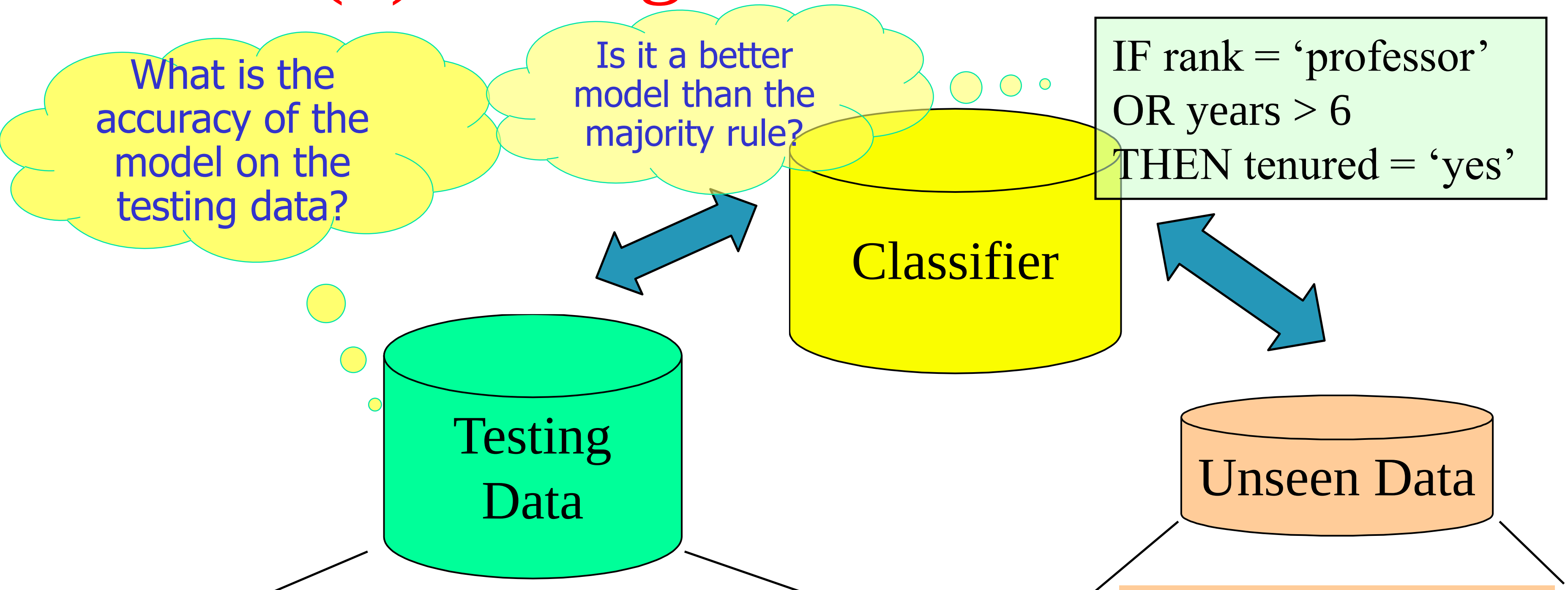
Classification Algorithms

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classifier (Model)

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Process (2): Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

(Jeff, Professor, 4)

Tenured?

Yes

Classification—Accuracy Estimation

- Estimate accuracy of the model
  - The known label is compared to the predicted label
  - Training Accuracy Rate = 1 - Err_{Tr}
    - The percentage of *training set* samples that are correctly classified by the model
  - Testing Accuracy Rate = 1 - Err_{Test}
    - The percentage of *test set* samples that are correctly classified by the model
    - Test set is independent of training set, otherwise over-fitting will occur
- If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Confidence Interval for an Error Rate

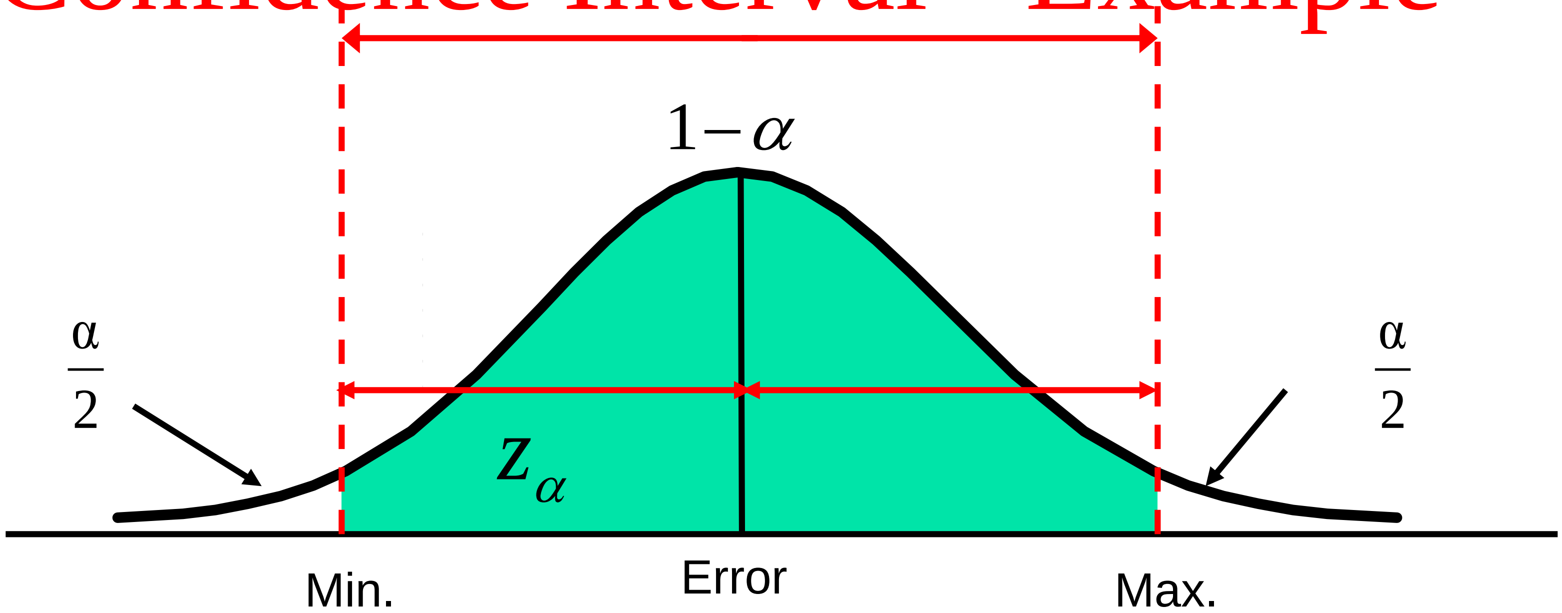
- Test error Err_{Test} is an *estimate* of the true error rate Err_{True} on the entire population
- Err_{Test} is governed by Binomial distribution approximated by Normal when $n \geq 30$
  - Assumption: *n* test samples are drawn randomly and independently from the entire population
- With the probability 1 - α, the true error rate Err_{True} lies in the *confidence interval*

How to estimate the true accuracy rate?

$$[Err_{Test} - z_{\alpha}\sqrt{\frac{Err_{Test}(1-Err_{Test})}{n}}; Err_{Test} + z_{\alpha}\sqrt{\frac{Err_{Test}(1-Err_{Test})}{n}}]$$

9

Confidence Interval - Example



z_α

alpha	0.2	0.1	0.05	0.01	0.001
Z_alpha	1.282	1.645	1.960	2.576	3.291

Error	n	alpha	z_alpha	min.	max.
0.200	30	0.010	2.576	0.0119	0.3881
0.200	30	0.050	1.960	0.0569	0.3431
0.200	30	0.100	1.645	0.0799	0.3201

10

Difference between Classifiers

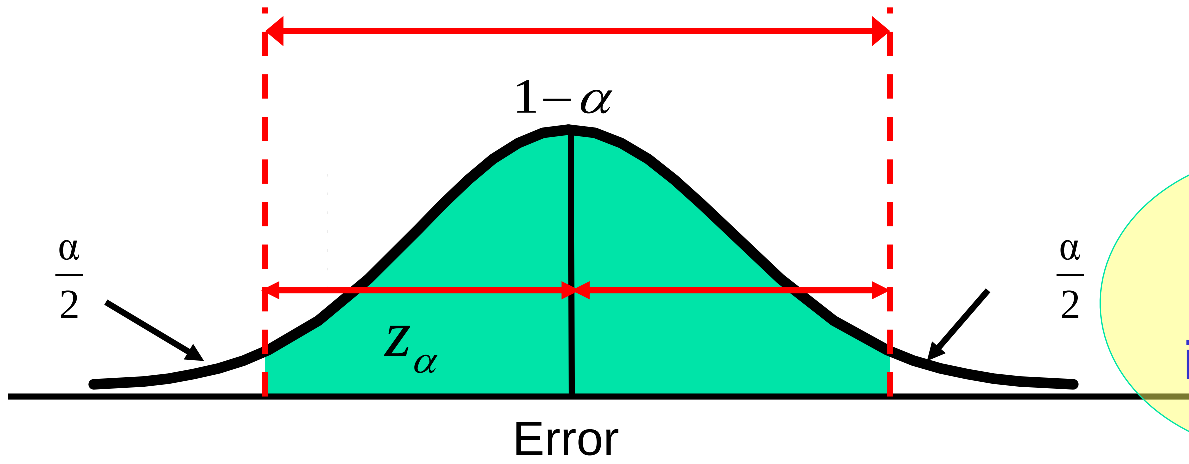
- Estimated difference between error rates

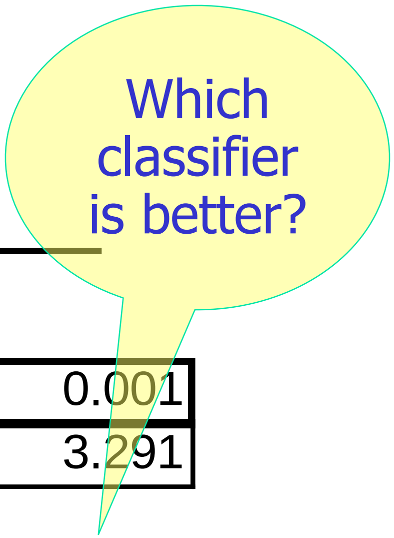
$$\hat{d} = Err_{Test1} - Err_{Test2}$$

- *d*^ is governed by Binomial distribution approximated by Normal when $n_1, n_2 \geq 30$
- With the probability 1 - α, the true difference *d* lies in the *confidence interval*

$$\hat{d} \pm z_{\alpha} \sqrt{\frac{Err_{Test1}(1 - Err_{Test1})}{n_1} + \frac{Err_{Test2}(1 - Err_{Test2})}{n_2}}$$

Difference between Classifiers – Example (Error1 vs. Error 2)





alpha	0.2	0.1	0.05	0.01	0.001
Z_alpha	1.282	1.645	1.960	2.576	3.291

Error1	n1	Error2	n2	d	alpha	z_alpha	min.	max.
0.200	30	0.400	40	0.20	0.010	2.326	-0.048	0.448
0.200	30	0.400	40	0.20	0.050	1.645	0.025	0.375
0.200	30	0.400	40	0.20	0.100	1.282	0.064	0.336

12

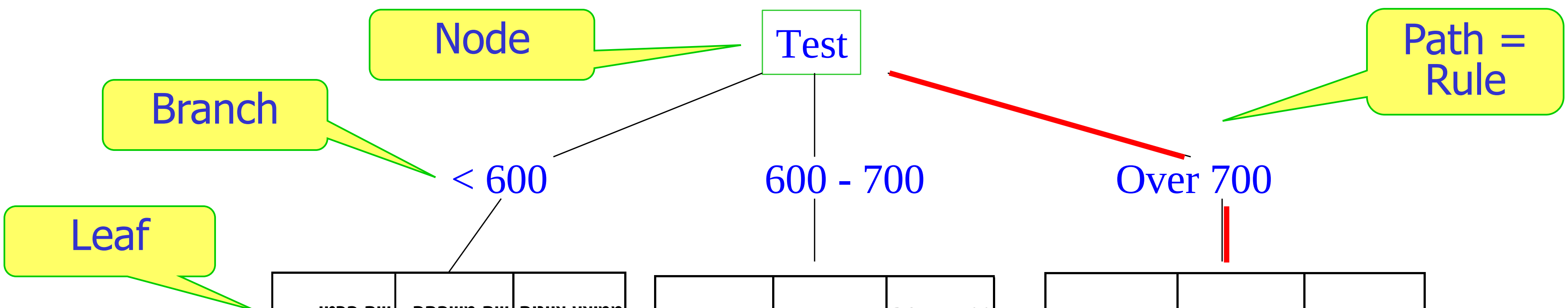
Lecture No. 5 – Decision Tree Learning

- Classification and Prediction
- Overview of Decision Tree Learning
- Avoiding Overfitting

Decision Tree Structure

Main Components

- *Nodes* - tests of some attribute
- *Branch* - one of possible values for the attribute
- *Leaves (terminal nodes)* - classifications
- *Path* (from the tree root to a leaf) - conjunction of attribute tests



שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Diana	Liberman	Medium
Anat	Klein	Low

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Ophir	Levy	Medium
Sharon	Grosman	High

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
David	Cohen	High

Decision Tree Learning

Appropriate Problems + Student Example

- Instances are described by a *fixed* set of attributes
  - Example: *Gender*, *Place of Birth*, and *Test Grade*
- Each predicting attribute takes a *small* number of disjoint possible values
  - Example: Place of Birth (Israel vs. Abroad)
- The target function has *discrete* output values (each value = class / concept)
  - Example: GPA (Low, Medium, High)
- *Disjunctive* rules are required
  - Example:
    - If (Test $<$ 600) Then GPA = Low
    - If (Test $\geq$ 600) Then GPA = Medium or High
- The training data may contain *errors* (noise)
- The training data may contain *missing attribute values*

Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm – e.g., ID3)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Detailed Example: Student Admission

All data in this example is fictional!

Historical (Training) Data:

ת"ז	שם פרטי	שם משפחה	מגדר	תאריך לידה	מקום לידה	שנת קבלה	ציון פסיכומטרי	שנת סיום	ממוצע ציונים
ID	**First Name**	**Last Name**	**Gender**	**Date of Birth**	**Place of Birth**	**Admission Year**	**Test Grade**	**Graduation Year**	**GPA**
543406619	David	Cohen	M	18/12/1979	USA	2002	730	2006	93.5
951984264	Ophir	Levy	M	11/07/1980	Israel	2002	680	2006	87.5
683168092	Sharon	Grosman	F	19/05/1981	Israel	2002	640	2006	94.3
100900927	Diana	Liberman	F	11/02/1980	Russia	2002	585	2006	85.8
516120403	Anat	Klein	F	03/02/1982	Israel	2002	570	2006	78.7

New (Scoring) Data:

?

ת"ז	שם פרטי	שם משפחה	מגדר	תאריך לידה	מקום לידה	שנת הגשת מועמדות	ציון פסיכומטרי	שנת סיום מתוכננת	ממוצע ציונים צפוי
ID	**First Name**	**Last Name**	**Gender**	**Date of Birth**	**Place of Birth**	**Application Year**	**Test Grade**	**Expected Graduation Year**	**Expected GPA**
537793401	Boaz	Bazak	M	22/09/1985	Israel	2007	580	2011	?
808943728	Ophir	Levy	M	10/02/1985	Israel	2007	650	2011	?
537362102	Maria	Neuman	F	03/12/1987	Ukraine	2007	720	2011	?

Pre-processing: Removing Irrelevant Features

Meaningless

Unique

Too many possible values

New value every year

New value every year

ת"ז ID	שם פרטי First Name	שם משפחה Last Name	מגדר Gender	תאריך לידה Date of Birth	מקום לידה Place of Birth	שנת קבלה Admission Year	ציון פסיכומטרי Test Grade	שנת סיום Graduation Year	ממוצע ציונים GPA
~~5434066[illegible]9~~	~~David~~	~~Cohen~~	M	~~18/12/1979~~	USA	~~2002~~	730	~~2006~~	93.5
~~951984264~~	~~Ophir~~	~~Levy~~	M	~~11/07/1980~~	Israel	~~2002~~	680	~~2006~~	87.5
~~683168092~~	~~Sharon~~	~~Grosman~~	F	~~10/05/1981~~	Israel	~~2002~~	640	~~2006~~	94.3
~~100900927~~	~~Diana~~	~~Liberman~~	F	~~11/02/1980~~	Russia	~~2002~~	585	~~2006~~	85.8
~~516120403~~	~~Anat~~	~~Klein~~	F	~~03/02/1982~~	Israel	~~2002~~	570	~~2006~~	78.7

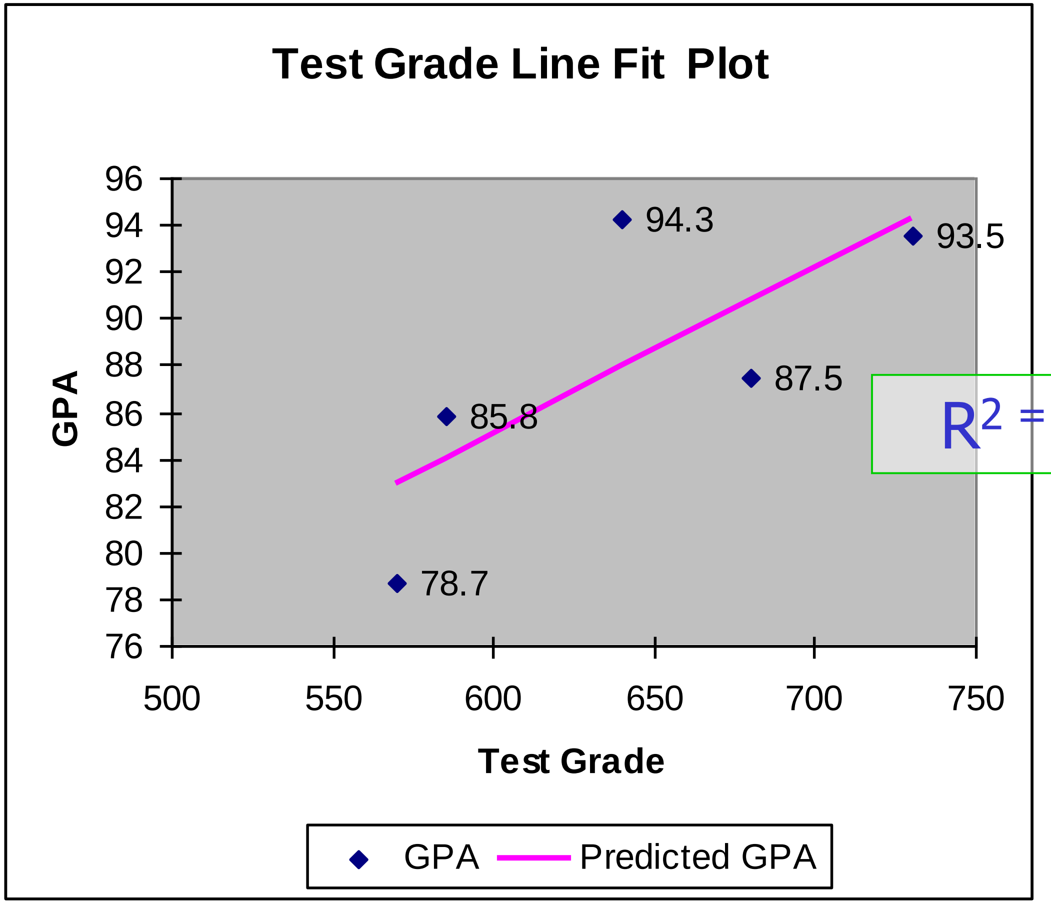
Remained features (attributes)

- Gender
- Place of Birth
- Test Grade

Predicted attribute: GPA

The problem: find the best (most accurate) model predicting GPA

Try 1:
Predict GPA Using a Linear Fit
Predictive attribute: Test Grade



$R^2 = 0.547$

More Pre-processing

Goal: reduce the number of values

Discretize to three intervals

Low	0-79
Medium	80-89
High	90-100

Generalize to two values (Israel vs. Diaspora)

Discretize to three intervals

שם פרטי	שם משפחה	מגדר	מקום לידה	ציון פסיכומטרי	ממוצע ציונים
First Name	**Last Name**	**Gender**	**Place of Birth**	**Test Grade**	**GPA**
David	Cohen	M	Diaspora	Over 700	High
Ophir	Levy	M	Israel	600-700	Medium
Sharon	Grosman	F	Israel	600-700	High
Diana	Liberman	F	Diaspora	0-600	Medium
Anat	Klein	F	[illegible]	0-600	Low
	Use:	[illegible]	[illegible]	[illegible]	**Target**

What is the accuracy of the majority rule?

Try 2: Predict GPA Using a *Tree*

Predictive attribute: Test Grade

Test

Accuracy = 50%

Accuracy = 50%

Accuracy = 100%

< 600

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Diana	Liberman	Medium
Anat	Klein	Low

Prediction = Medium or Low (50%/50%)

600 - 700

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Ophir	Levy	Medium
Sharon	Grosman	High

Prediction = High or Medium (50%/50%)

Over 700

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
David	Cohen	High

n = High

Is it a better model than the majority rule?

Average accuracy: 0.4*50% + 0.4*50% + 0.2*100% = 60%

Try 2: Predict GPA Using a *Tree*

Predictive attribute: Gender

Accuracy = 33%

Gender

Accuracy = 50%

F

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Sharon	Grosman	High
Diana	Liberman	Medium
Anat	Klein	Low

Prediction = High or Medium or Low (33%/33%/33%)

M

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
David	Cohen	High
[illegible]	Levy	Medium

Is it a better model than the majority rule?

[illegible]tion = High or Medium (50%/50%)

Average accuracy: 0.4*50% + 0.6*33% = 40%

Try 2: Predict GPA Using a *Tree*

Predictive attribute: Place of Birth

Place of Birth

Accuracy = 33%

Israel

שם פרטי First Name	שם משפחה Last Name	ממוצע ציונים GPA
Ophir	Levy	Medium
Sharon	Grosman	High
Anat	Klein	Low

Prediction = High or Medium or Low (33%/33%/33%)

Accuracy = 50%

Diaspora

שם פרטי First Name	שם משפחה Last Name	ממוצע ציונים GPA
David	Cohen	High
Diana	Liberman	Medium

Prediction = High or Medium (50%/50%)

Is it a better model than the majority rule?

Average accuracy: 0.6*33% + 0.4*50% = 40%

How to choose the best tree?

Test

< 600

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Diana	Liberman	Medium
Anat	Klein	Low

600 - 700

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Ophir	Levy	Medium
Sharon	Grosman	High

Over 700

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
David	Cohen	High

Training accuracy:
0.4*50% +
0.4*50% +
0.2*100% = 60%

Gender

F

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Sharon	Grosman	High
Diana	Liberman	Medium
Anat	Klein	Low

M

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
David	Cohen	High
Ophir	Levy	Medium

Training accuracy:
0.4*50% + 0.6*33%
= 40%

Place of Birth

Israel

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
Ophir	Levy	Medium
Sharon	Grosman	High
Anat	Klein	Low

Diaspora

שם פרטי	שם משפחה	ממוצע ציונים
First Name	**Last Name**	**GPA**
David	Cohen	High
Diana	Liberman	Medium

Training accuracy:
0.6*33% + 0.4*50% =
40%

Test, Gender or Place of Birth?

ID3: Use Information Gain

Attribute Selection Measure in ID3: Information Gain

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in *D* belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in *D*:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information (conditional entropy) needed (after using *A* to split *D* into v partitions) to classify *D*:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

- Information gained (mutual information) by branching on attribute *A*

$$Gain(A) = Info(D) - Info_A(D)$$

Student Admission Example

Expected information (entropy) needed to classify a tuple in *D*:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Number of classes (*m*): 3

The classes:

- **Low** (one example)
- **Medium** (two examples)
- **High** (two examples)

Low	1
p	0.200
-logp	2.322
Medium	2
p	0.400
-logp	1.322
High	2
p	0.400
-logp	1.322
Total	5
p	1.00
Entropy	**1.522**

0.2*2.322 + 0.4*1.322 + 0.4*1.322 = 1.522

Student Admission Example (cont.)

Attribute:

Test Grade

Values:
- 0-600
- 600-700
- Over 700

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

$$Info(D) = 1.522$$

	Test Grade			**Total**
	0-600	600-700	Over 700	
Low	1	0	0	1
p	0.500	0.000	0.000	
-logp	1.000	0.000	0.000	
Medium	1	1	0	2
p	0.500	0.500	0.000	
-logp	1.000	1.000	0.000	
High	0	1	1	2
p	0.000	0.500	1.000	
-logp	0.000	1.000	0.000	
Total	2	2	1	5
p	0.40	0.40	0.20	1.00
Entropy	1.000	1.000	0.000	0.800
Gain				**0.722**

0.5*1.0 + 0.5*1.0 = 1.0   0.4*1.0 + 0.4*1.0 + 0.2*0.0 = 0.8

Student Admission Example (cont.)

Attribute:

Gender

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

$$Info(D) = 1.522$$

	Gender		Total
	M	F	
Low	0	1	1
p	0.000	0.333	
-logp	0.000	1.585	
Medium	1	1	2
p	0.500	0.333	
-logp	1.000	1.585	
High	1	1	2
p	0.500	0.333	
-logp	1.000	1.585	
Total	2	3	5
p	0.40	0.60	1.00
Entropy	1.000	1.585	1.351
Gain			**0.171**

Student Admission Example (cont.)

Attribute:

Place of Birth

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j)$$

$$Gain(A) = Info(D) - Info_A(D)$$

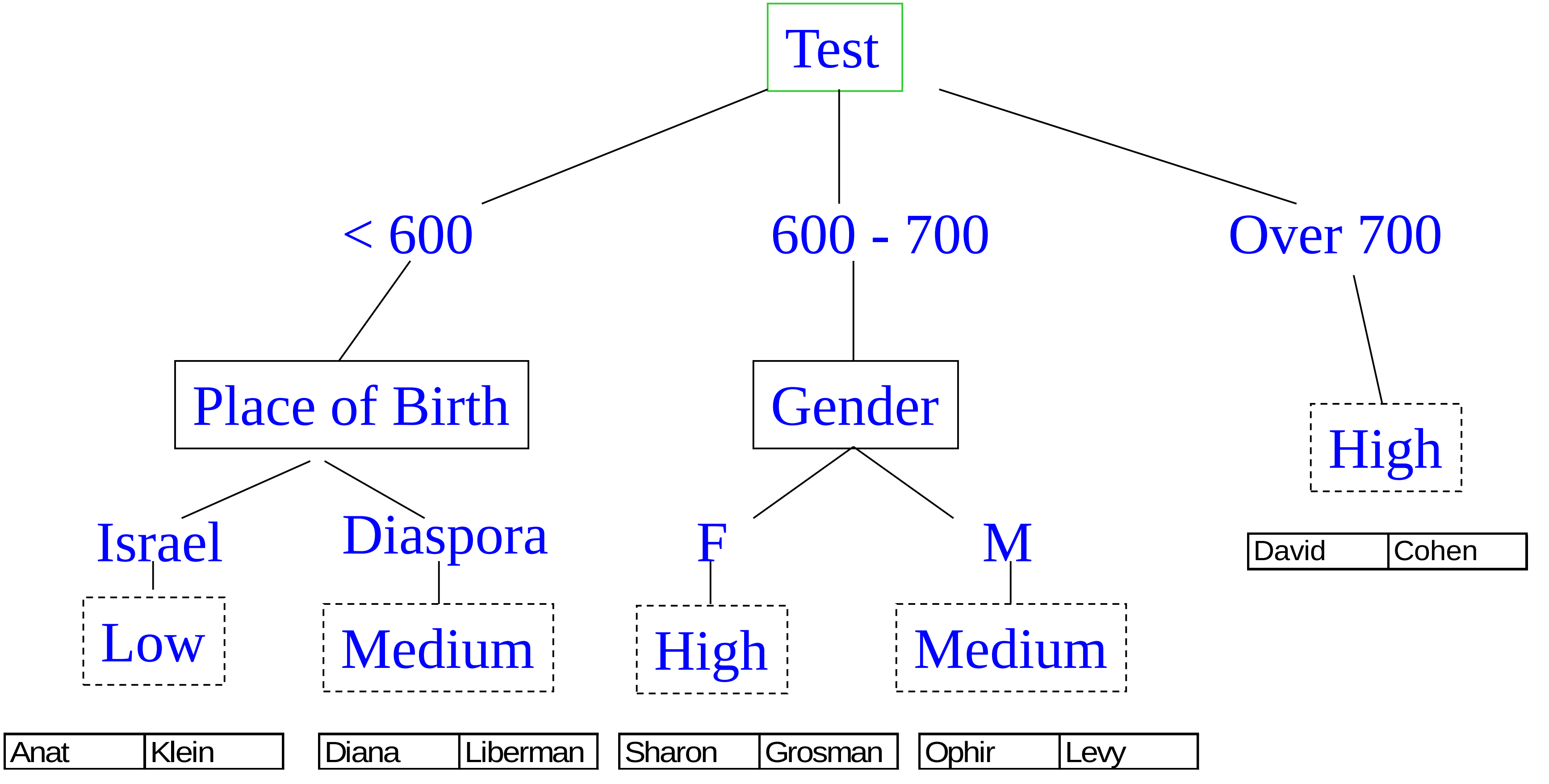
$$Info(D) = 1.522$$

	Place of Birth		Total
	Israel	Abroad	
Low	1	0	1
p	0.333	0.000	
-logp	1.585	0.000	
Medium	1	1	2
p	0.333	0.500	
-logp	1.585	1.000	
High	1	1	2
p	0.333	0.500	
-logp	1.585	1.000	
Total	3	2	5
p	0.60	0.40	1.00
Entropy	1.585	1.000	**1.351**
Gain			**0.171**

Student Admission Example (cont.)

- Gain (*Test Grade*) = 0.722
- Gain (*Gender*) = 0.171
- Gain (*Place of Birth*) = 0.171
- Selected attribute: **Test Grade**

Student Admission Example Complete Decision Tree



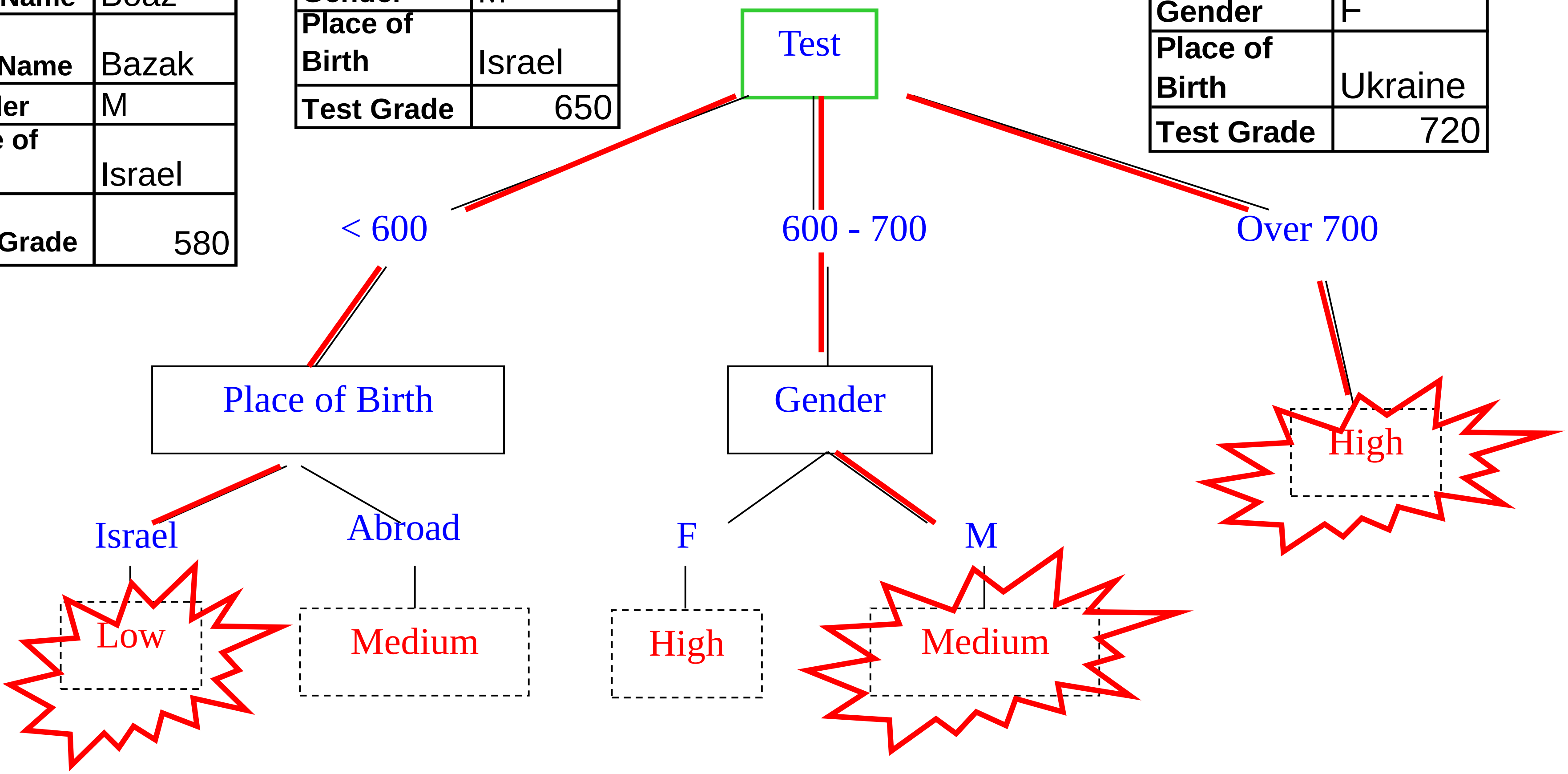
Training accuracy: 100%

Student Admission Example
Classification with Decision Tree

First Name	Boaz
Last Name	Bazak
Gender	M
Place of Birth	Israel
Test Grade	580

First Name	Ophir
Last Name	Levy
Gender	M
Place of Birth	Israel
Test Grade	650

First Name	Maria
Last Name	Neuman
Gender	F
Place of Birth	Ukraine
Test Grade	720



Lecture No. 5 – Decision Tree Learning

- Classification and Prediction
- Overview of Decision Tree Learning
- Avoiding Overfitting

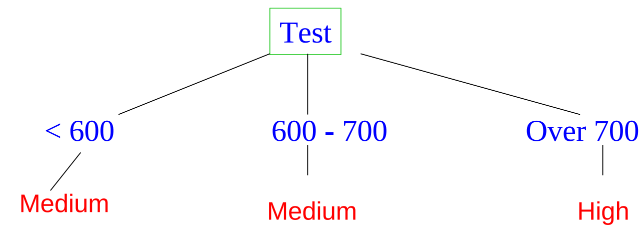
Overfitting

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Definition of overfitting
  - *h* – hypothesis (e.g., decision tree) in a *hypothesis space H*
  - Training data: $error_{train}(h)$
  - Entire population *D*: $error_D(h)$
  - Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that
    - $error_{train}(h) < error_{train}(h')$ and
    - $error_D(h) > error_D(h')$

Overfitting – Student Example

Tree1 (h'):

ID	First Name	Last Name	Expected GPA	Actual GPA	Error
537793401	Boaz	Bazak	**Medium**	Medium	**No**
808943728	Ophir	Levy	**Medium**	Medium	**No**
537362102	Maria	Neuman	**High**	High	**No**



$error_D\,(h') = 0\%$

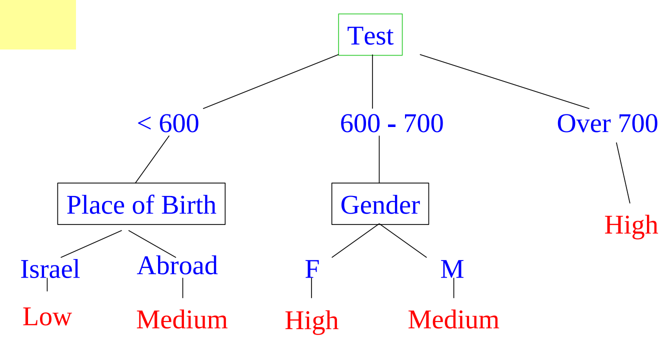
$error_{train}\,(h') = 40\%$

$error_{train}\,(h) < error_{train}\,(h')$

$error_D\,(h) > error_D\,(h')$

Tree2 (h):

ID	First Name	Last Name	Expected GPA	Actual GPA	Error
537793401	Boaz	Bazak	**Low**	Medium	**Yes**
808943728	Ophir	Levy	**Medium**	Medium	**No**
537362102	Maria	Neuman	**High**	High	**No**

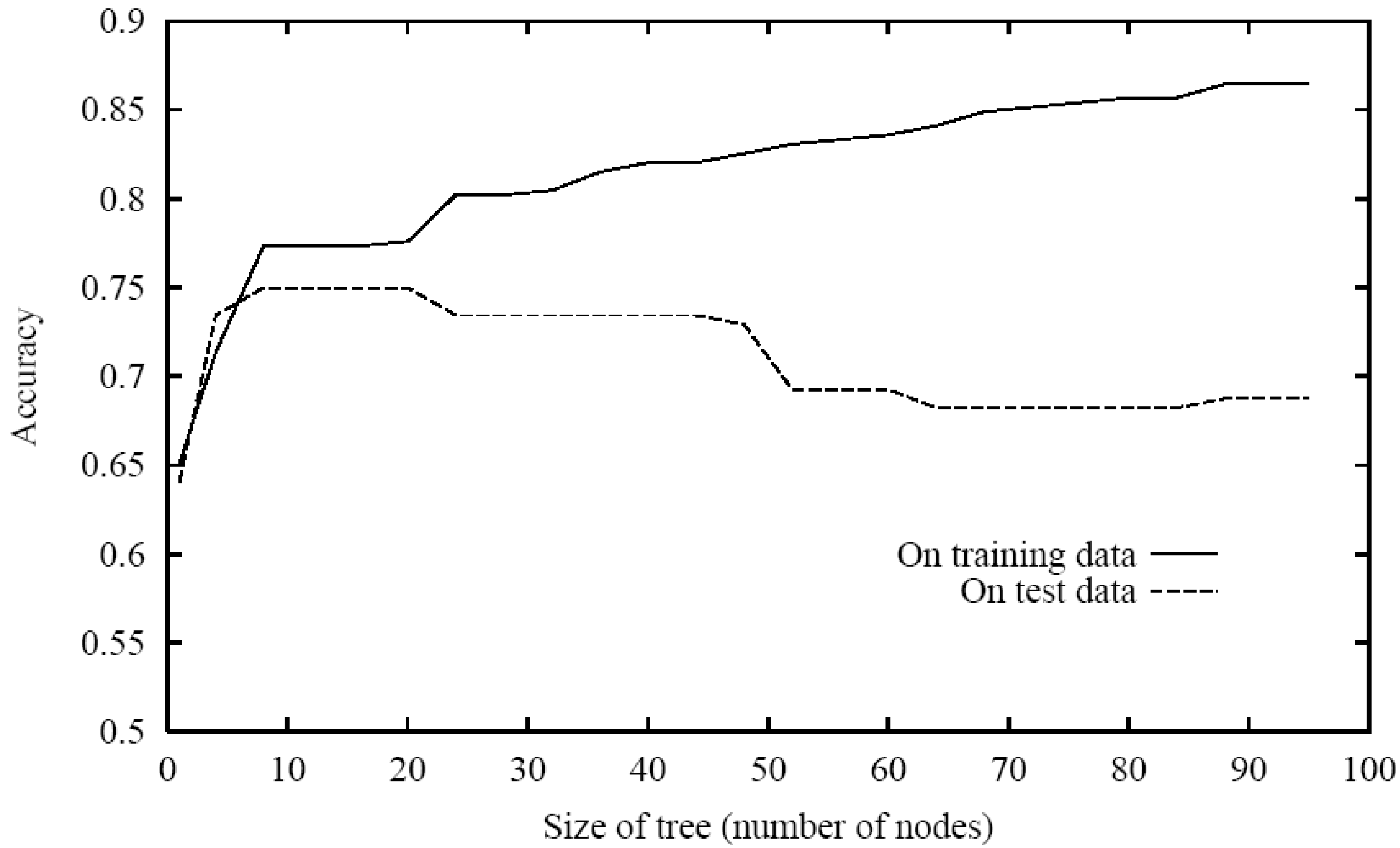


$error_D\,(h) = 33\%$

Which tree is better?

$error_{train}\,(h) = 0\%$

Overfitting in Decision Tree Learning
(Source: Mitchell, 1997)



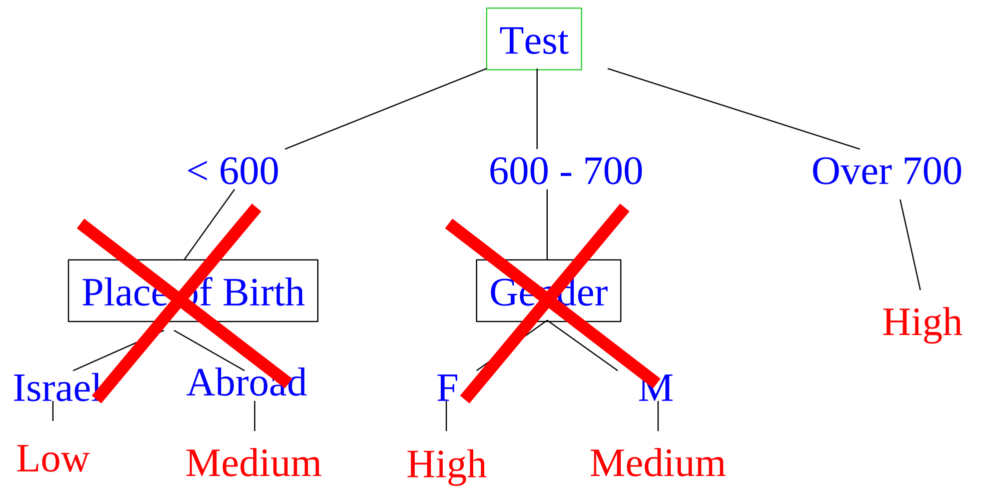
Avoiding Overfitting

- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early —do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

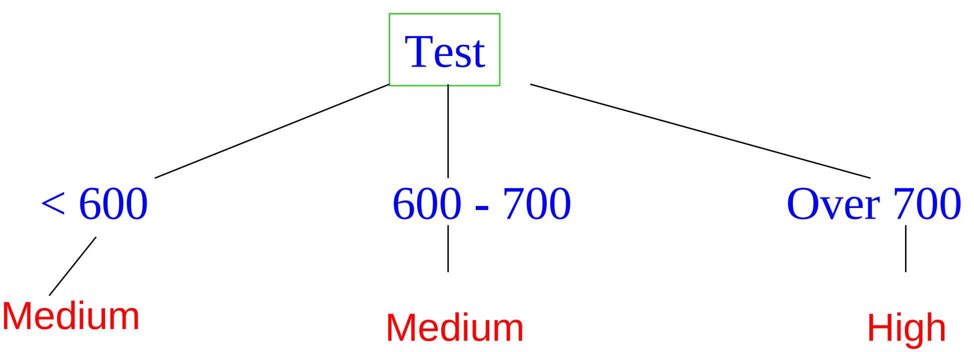
Avoiding Overfitting - Student Example

"Fully grown" tree:

Prune two nodes:



Pruned tree:



38

Approaches to Avoid Overfitting and Determine the Final Tree Size

- Separate training (2/3) and testing (1/3) sets
- Use cross validation, e.g., 10-fold cross validation
- Use all the data for training
  - but apply a statistical test (e.g., chi-square) to estimate whether expanding or pruning a node may improve the entire distribution
- Use minimum description length (MDL) principle
  - halting growth of the tree when the encoding is minimized

Chi-Square Test

(based on Quinlan, Induction of Decision Trees, 1986)

- Notation
  - A - splitting (branching) attribute (e.g., Test)
  - v - domain size of attribute A (3: 0-600, 600-700, 700+)
  - C_i - subset of records containing value i of attribute A (Test < 600: 2)
  - c – number of classes (3: low, medium, high)
  - e_j - number of records belonging to class j in the entire data set C (*Low* = 1, *Medium* = 2, *High* = 2)
  - o_{ij} - number of records belonging to class j in subset C_i (Test < 600, GPA = Low : 1)
  - α - significance level

Chi-Square Test (cont.)

- **Null Hypothesis**: attribute *A* is irrelevant to classifying the records in data set *C*
- **Alternative Hypothesis**: attribute *A* affects the class distribution in data set *C*
- Expected number of class *j* records in subset C_i:
- e_j – actual number of records in class *j*
- o_{ij} - actual number of class *j* records in subset C_i

$$e'_{ij} = \frac{e_j}{\sum_{j=1}^{c} e_j} \sum_{j=1}^{c} o_{ij}$$

Statistic:

$$\sum_{j=1}^{c} \sum_{i=1}^{v} \frac{(o_{ij} - e'_{ij})^2}{e'_{ij}} \sim \chi_{\alpha}^{2}((v-1)(c-1))$$

v - domain size of *A*

c – number of classes

Chi-Square Test – Student Example

$$e'_{ij} = \frac{e_j}{\sum_{j=1}^{c} e_j} \sum_{j=1}^{c} o_{ij}$$

- Entire data set (before splitting): $e_{Low} = 1$, $e_{Medium} = 2$, $e_{High} = 2$
- Splitting by *Test*
  - Test < 600: *Low = 1, Medium = 1* $\sum_{j=1}^{c} o_{ij} = 2$
    - e'_{low}= (1/5)*2 = 0.4, e'_{medium} = (2/5)*2 = 0.8. e'_{high} = (2/5)*2 = 0.8.
  - Test = 600-700: *Medium = 1, High = 1* $\sum_{j=1}^{c} o_{ij} = 2$
    - e'_{low}= (1/5)*2 = 0.4, e'_{medium} = (2/5)*2 = 0.8. e'_{high} = (2/5)*2 = 0.8.
  - Test > 700: *High = 1* $\sum_{j=1}^{c} o_{ij} = 1$
    - e'_{low}= (1/5)*1 = 0.2, e'_{medium} = (2/5)*1 = 0.4. e'_{high} = (2/5)*1 = 0.4.

Chi-Square Test – Student Example (cont.)

		Test Grade (i)			Total	p_j
GPA (j)		0-600	600-700	Over 700		
Low	Actual	1	0	0	1	0.2
	Expected	**0.4**	**0.4**	**0.2**	1	
	Statistic	**0.9**	**0.4**	**0.2**	**1.5**	
Medium	Actual	1	1	0	2	0.4
	Expected	**0.8**	**0.8**	**0.4**	2	
	Statistic	**0.05**	**0.05**	**0.4**	**0.5**	
High	Actual	0	1	1	2	0.4
	Expected	**0.8**	**0.8**	**0.4**	2	
	Statistic	**0.8**	**0.05**	**0.9**	**1.75**	
Total		2	2	1	**3.75**	5

Statistic: $\boxed{\sum_{j=1}^{c} \sum_{i=1}^{v} \frac{(o_{ij} - e'_{ij})^2}{e'_{ij}} = 3.75}$ $\boxed{\chi^2_{0.05}(4) = 9.49}$

Conclusion: do not split the node on *Test Grade*

Pessimistic Error Pruning (PEP)

- Uses training set to estimate error on new data
- Error estimate (relative frequency with continuity correction)
  - probability of error (apparent error rate)

$$q = \frac{N - n_C + 0.5}{N}$$

  - where
    - N = #examples
    - n_C = #examples in majority class

Pessimistic Error Pruning (cont.)

- Error of a node *v* (if pruned)
  - where
    - N_v = #examples at node *v*
    - $n_{C,v}$ = #examples in majority class at node *v*

$$q(v) = \frac{N_v - n_{C,v} + 0.5}{N_v}$$

- Error of a subtree *T*
  - Where
    - l = leaf node of sub-tree *T*

$$q(T) = \frac{\sum_{l \in leafs(T)} (N_l - n_{C,l} + 0.5)}{\sum_{l \in leafs(T)} N_l}$$

- Prune if $q(v) \leq q(T)$
- Prunes in bottom-up fashion
  - fast
  - considered a weakness (on accuracy)

Example of Post-Pruning

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

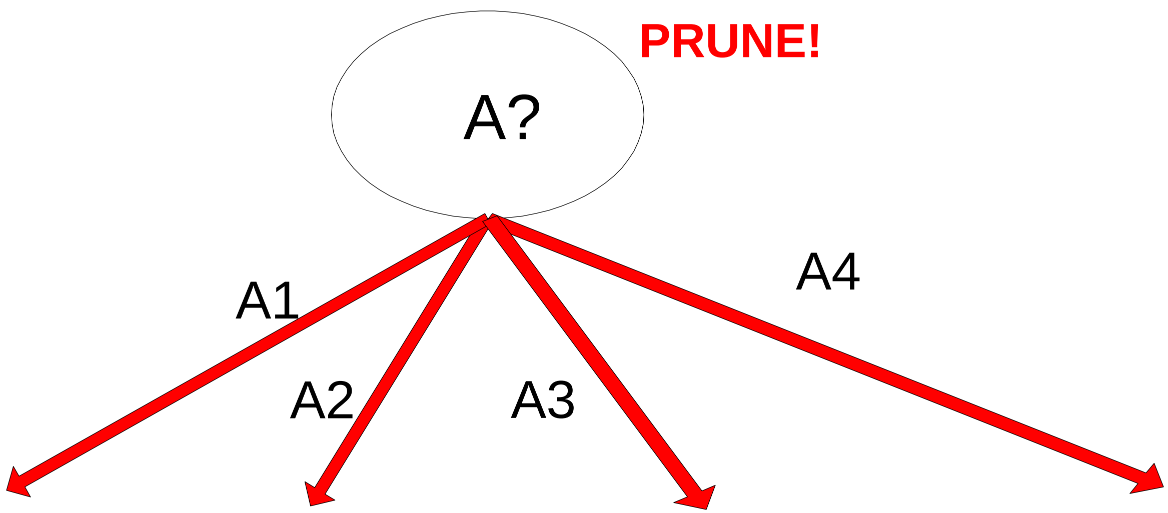
Training Error (Before splitting) = 10/30

Pessimistic error = *q(v)* = (10 + 0.5)/30 = 10.5/30

Training Error (After splitting) = 9/30

Pessimistic error (After splitting) *q(T)* = (9 + 4 × 0.5)/30 = 11/30

PRUNE!



Class = Yes	8
Class = No	4

Class = Yes	3
Class = No	4

Class = Yes	4
Class = No	1

Class = Yes	5
Class = No	1