Decision Tree

Machine learning 2021

Mansoor Rezghi

Ref:

Zaki

Data mining Tan

Supervised
- Regression
- Classification

Bayes - Naive - kNN

Regression Based
- SVM
- Perceptron

lin Reg

Logistic Reg

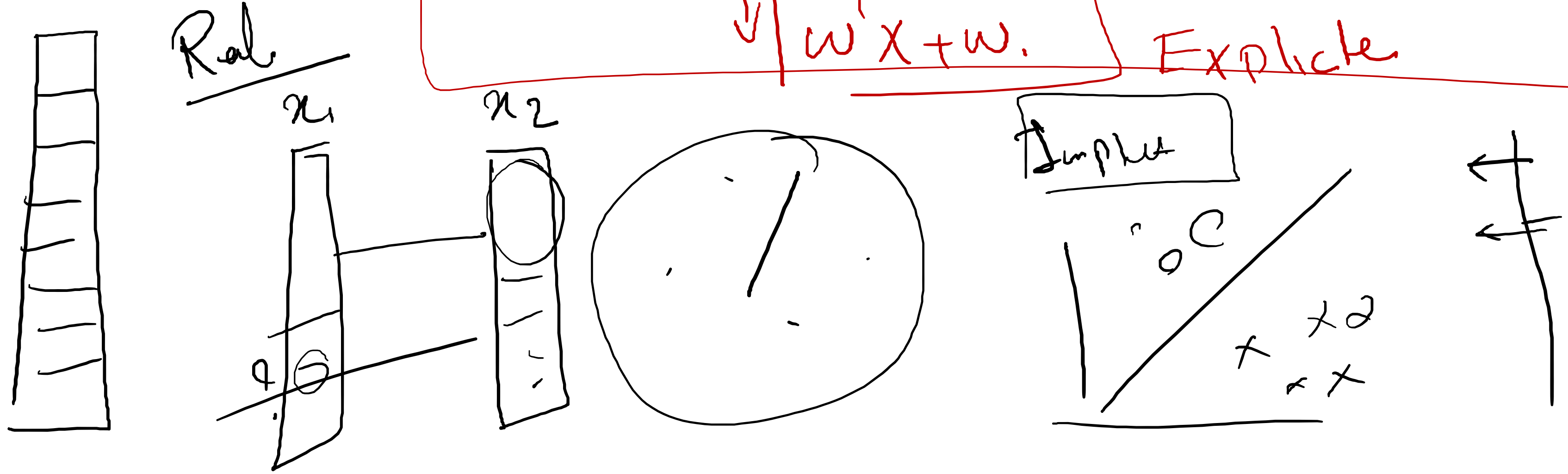
$y = f(x)$

$w^T x + w_0$

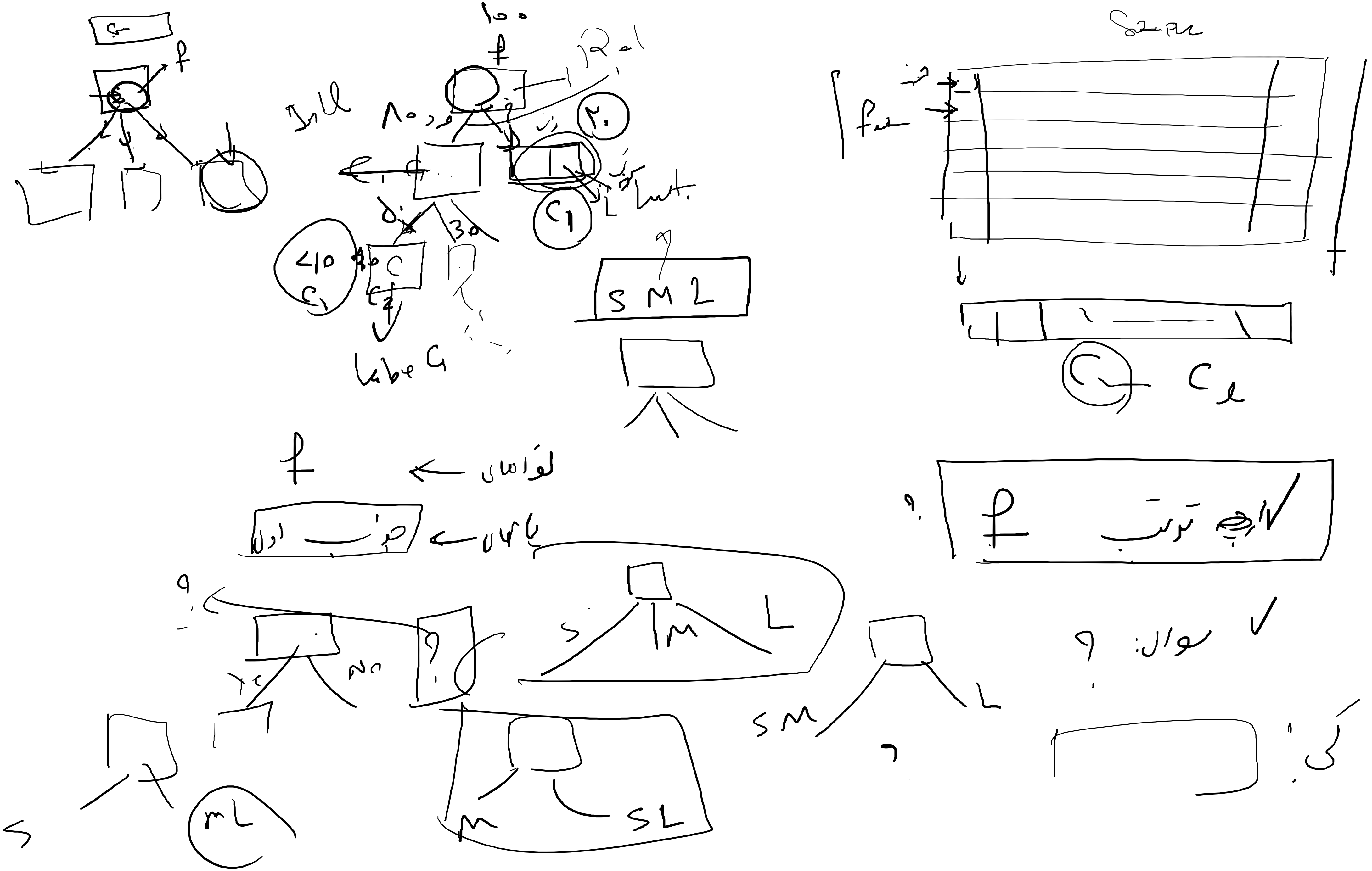
Explicit

knn → nonlin

lin



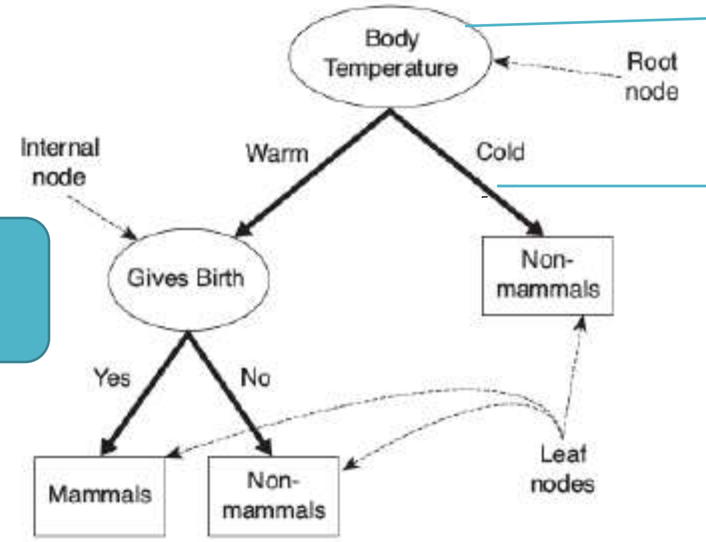
Di



SML
SM
L
SL
mL
سوال: ؟

Decision Tree classifiers

- A **root node** that has no incoming edges and zero or more outgoing edges.
- **Internal nodes**, each of which has exactly one incoming edge and two or more outg
- **Leaf** or **tern** and no outgoi



Questions

Answer

Implicit discrimination

Figure 4.4. A decision tree for the mammal classification problem.

S M D

Depth

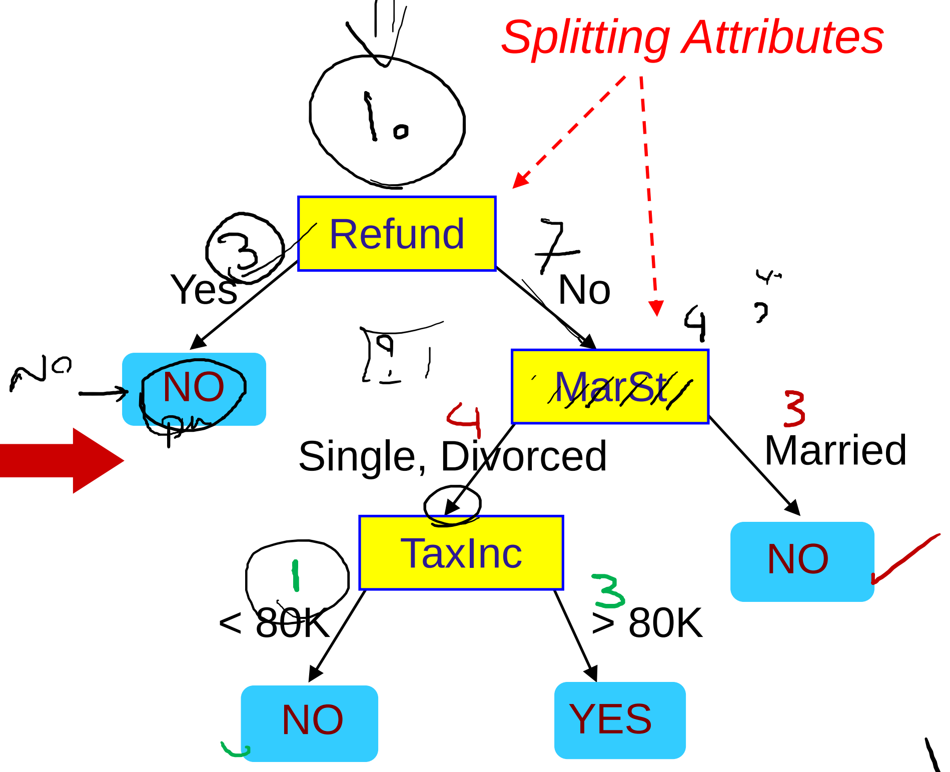
Example of a Decision Tree

categorical categorical continuous class

Train

Tid	Refund	Marital Status	Taxable Income	Cheat
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

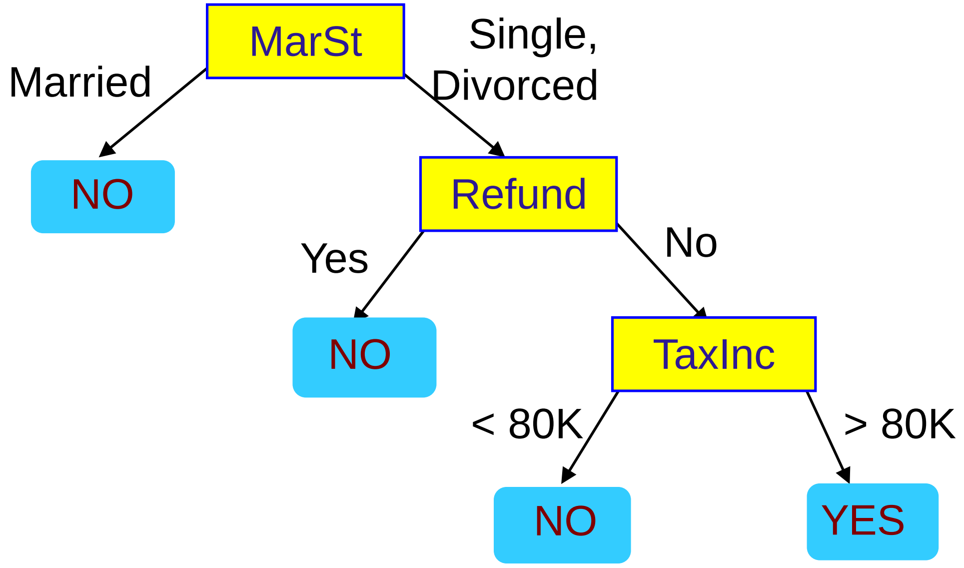


Model: Decision Tree

Another Example of Decision Tree

categorical categorical continuous class

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
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**

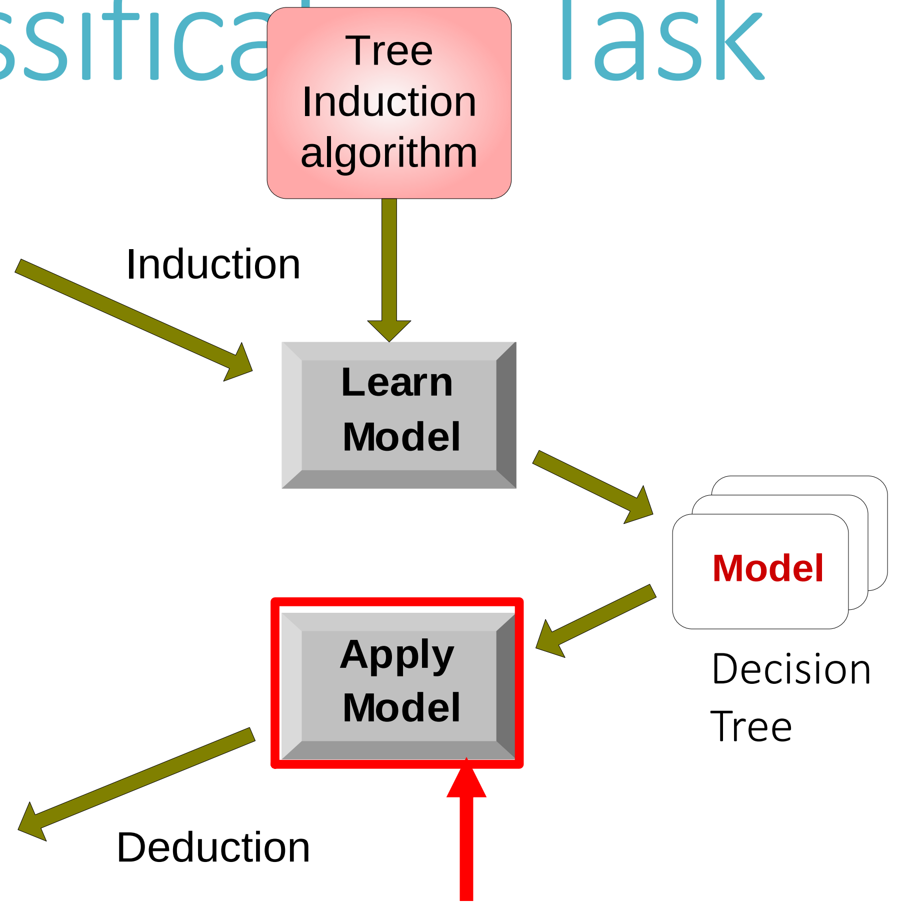


There could be more than one tree that fits the same data!

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

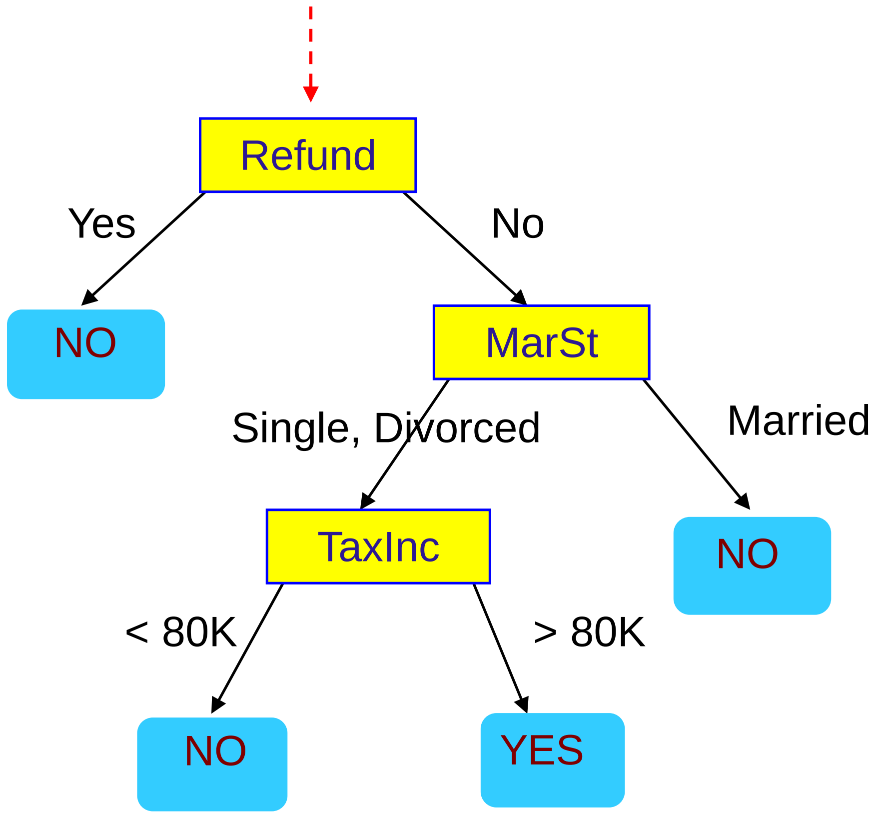


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Apply Model to Test Data

Start from the root of tree.



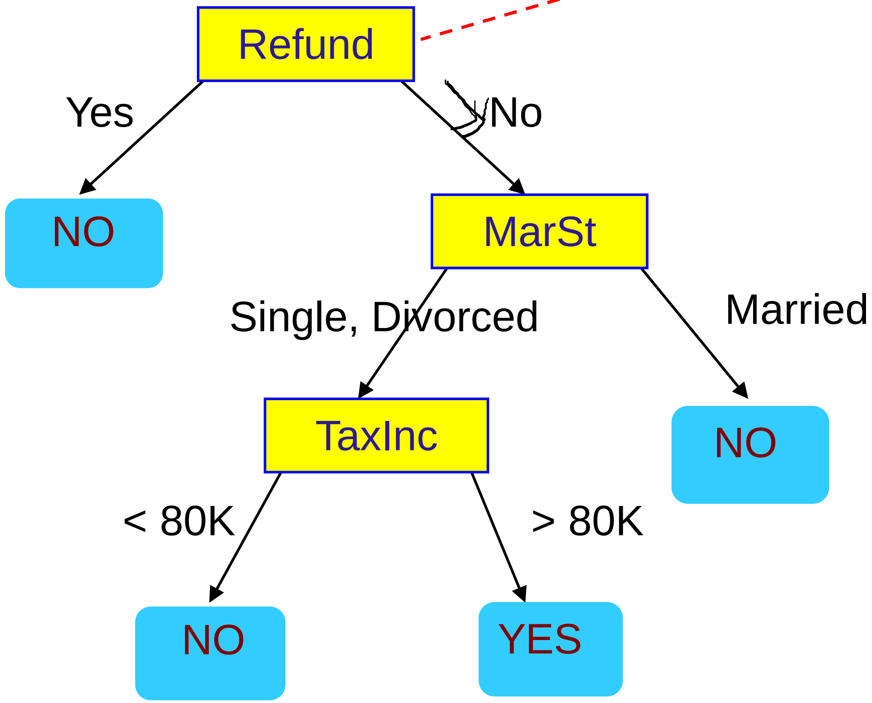
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	**?**

Apply Model to Test Data

Test Data

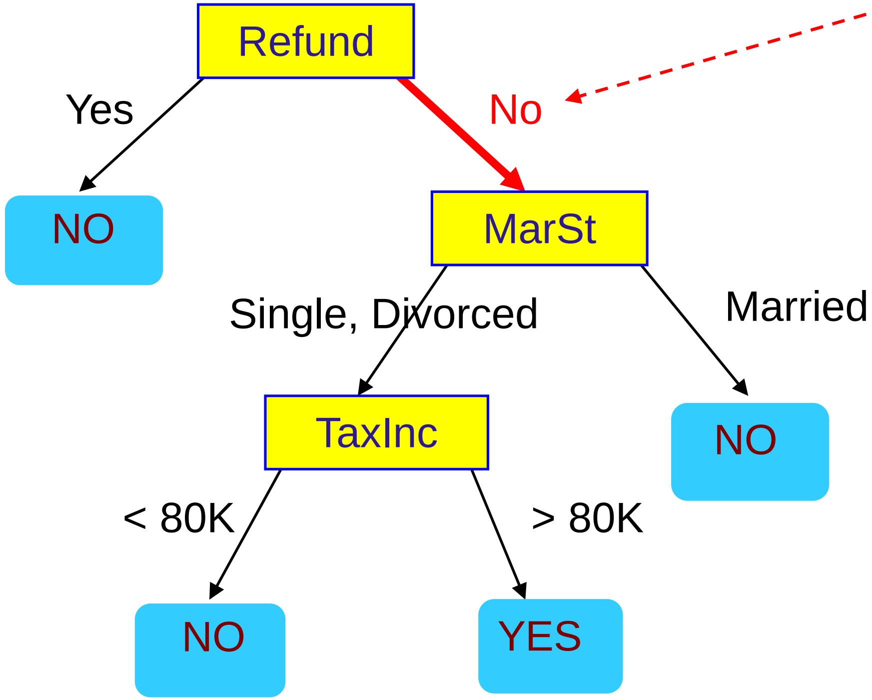
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

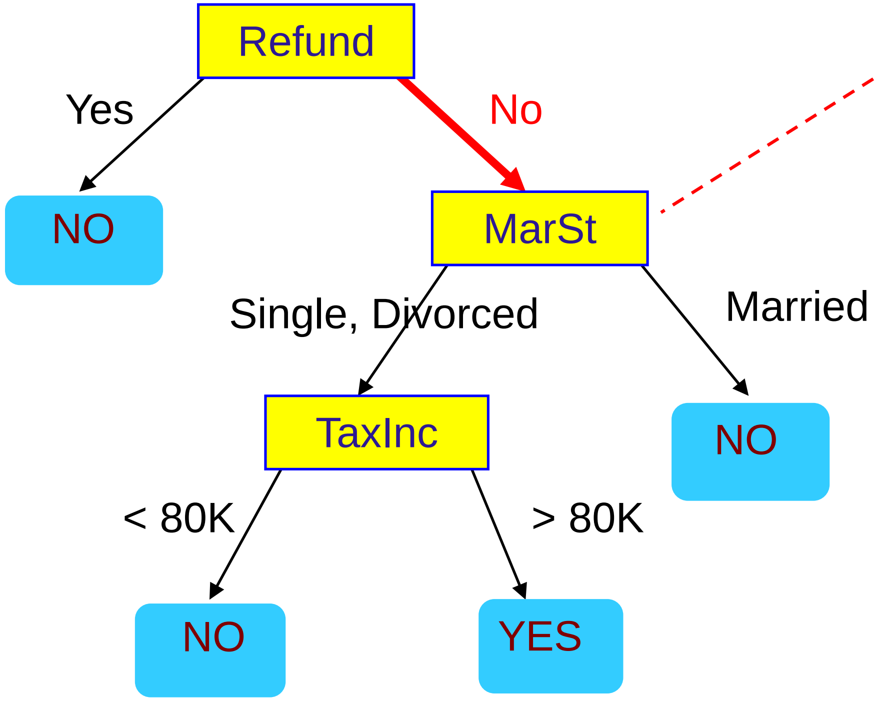
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

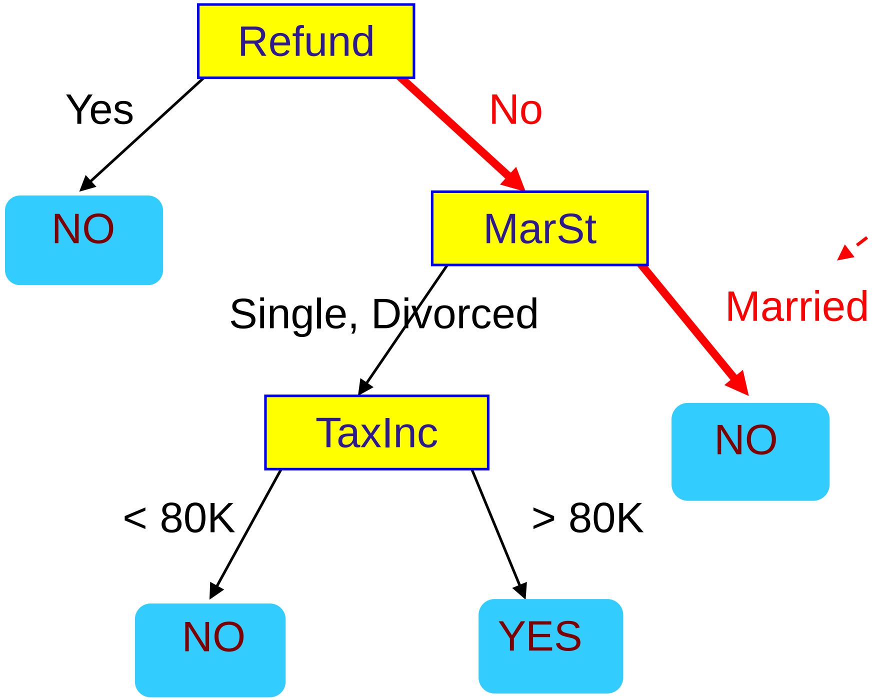
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



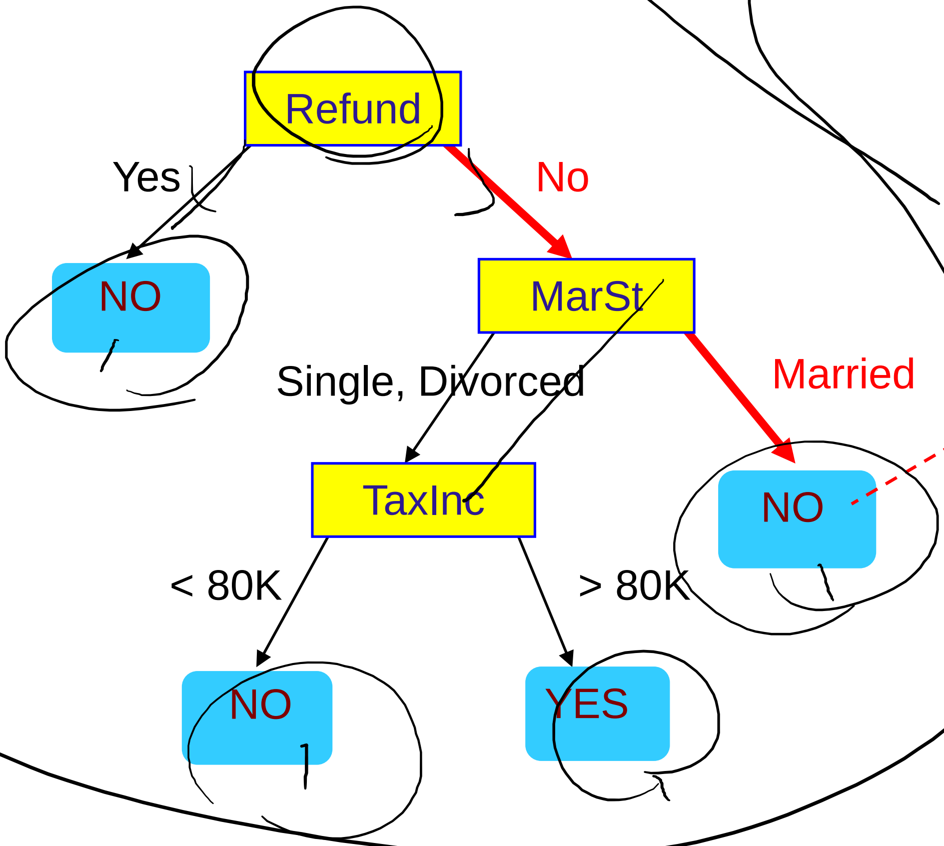
Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data



Test Data

X Y

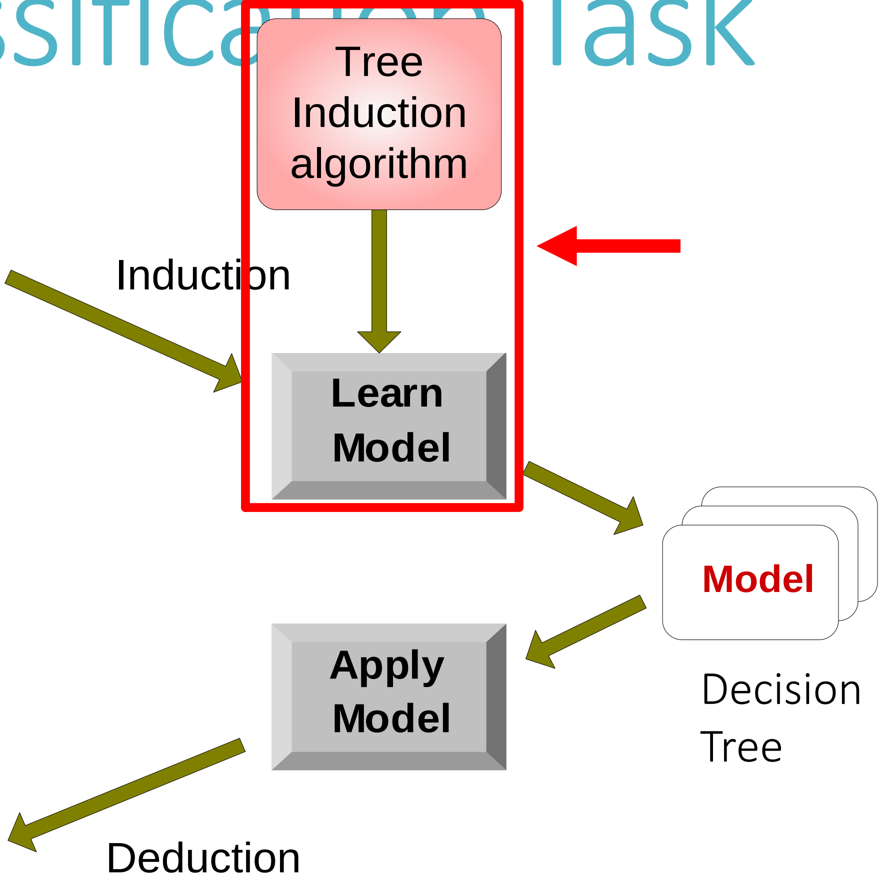
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	? NO

Assign Cheat to "No"

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



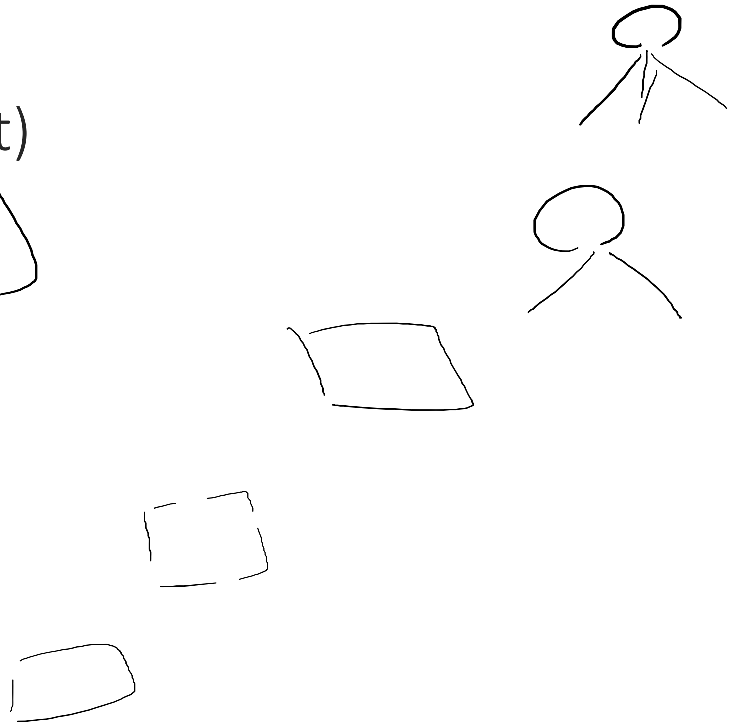
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Decision Tree Induction

Many Algorithms:

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



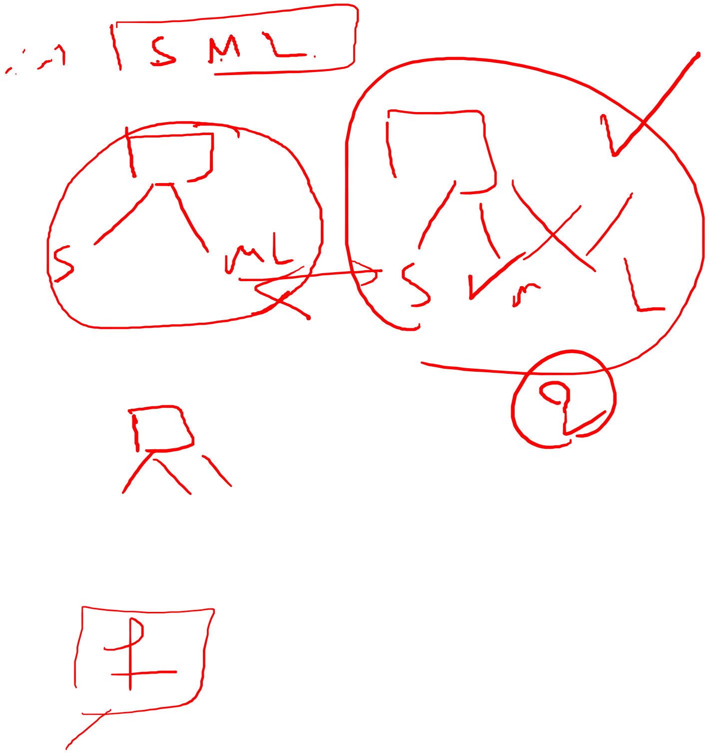
Design Issues of Decision Tree Induction

Design Issues of Decision Tree Induction

- Order of selecting the attributes (Evaluation)
- Designing the Questions (Evaluation)

How should the splitting procedure stop?

- Under and over fitting!
- Evaluation is need



Zaki

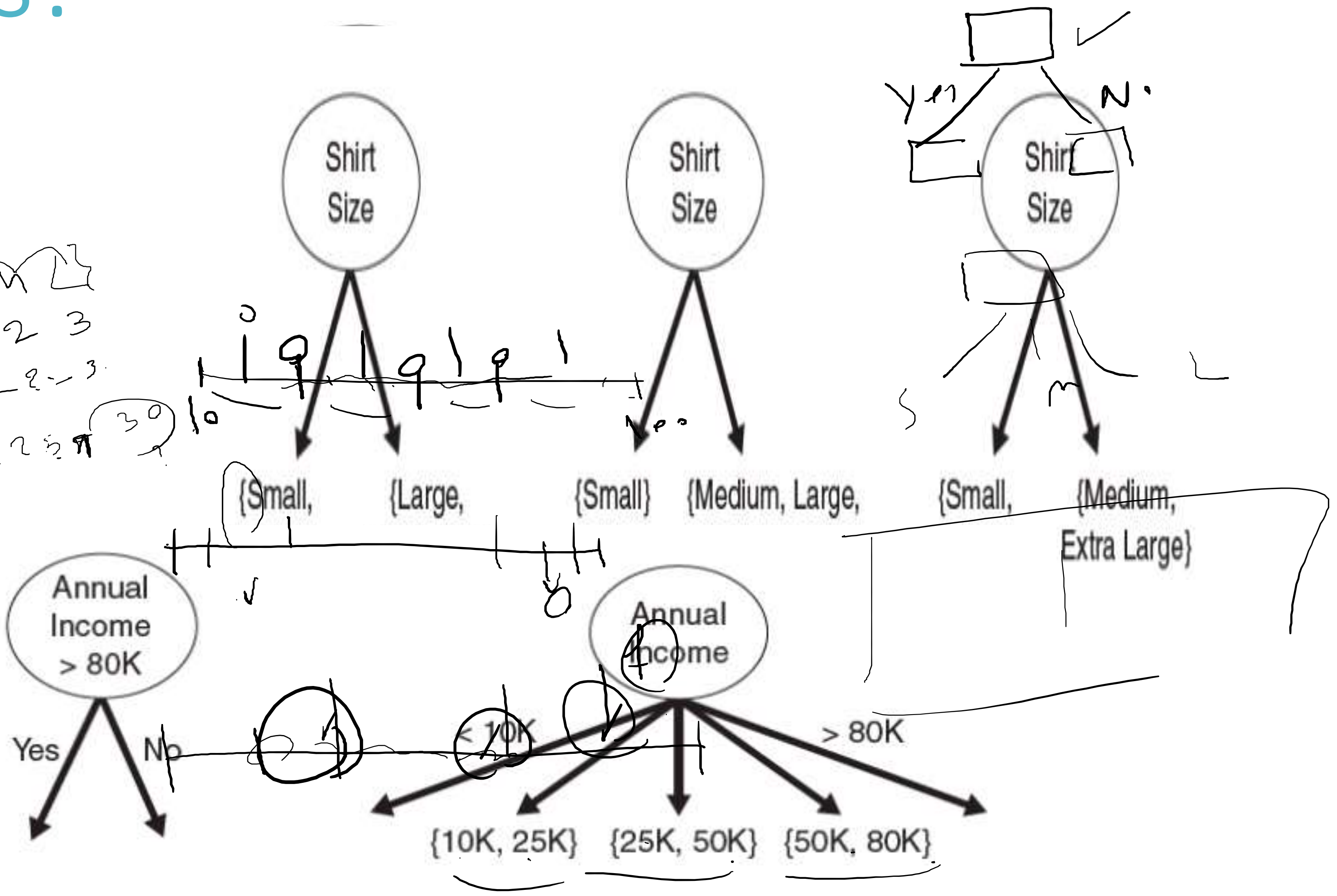
Tan Data Mng.

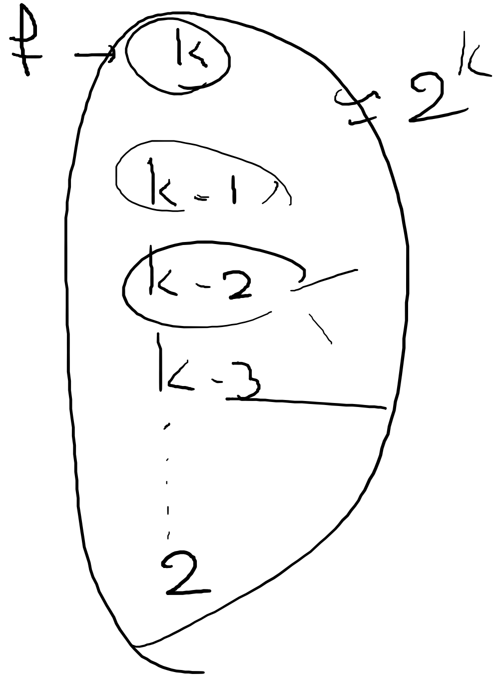
Questions?

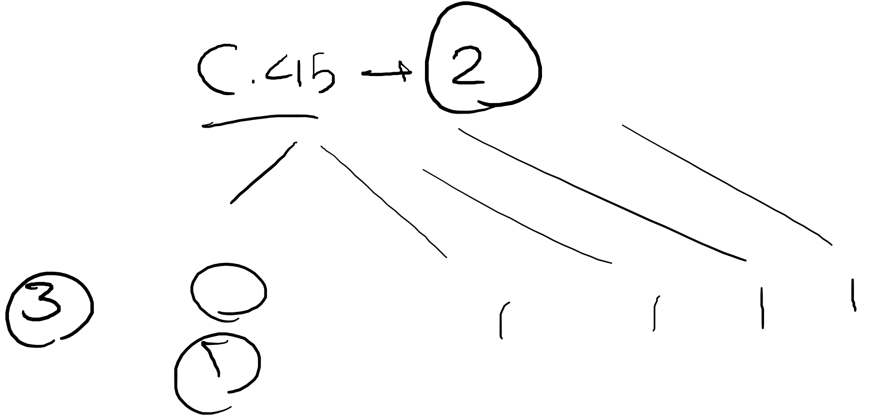
- Binary
- Nominal
- Ordinal
- Continuous

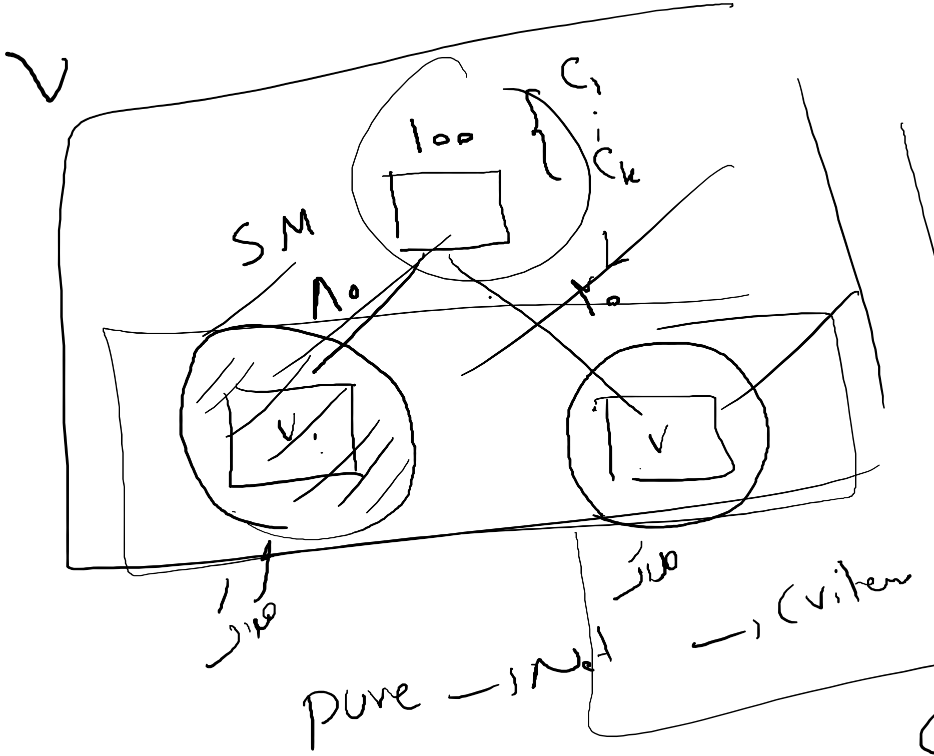
{S M L}

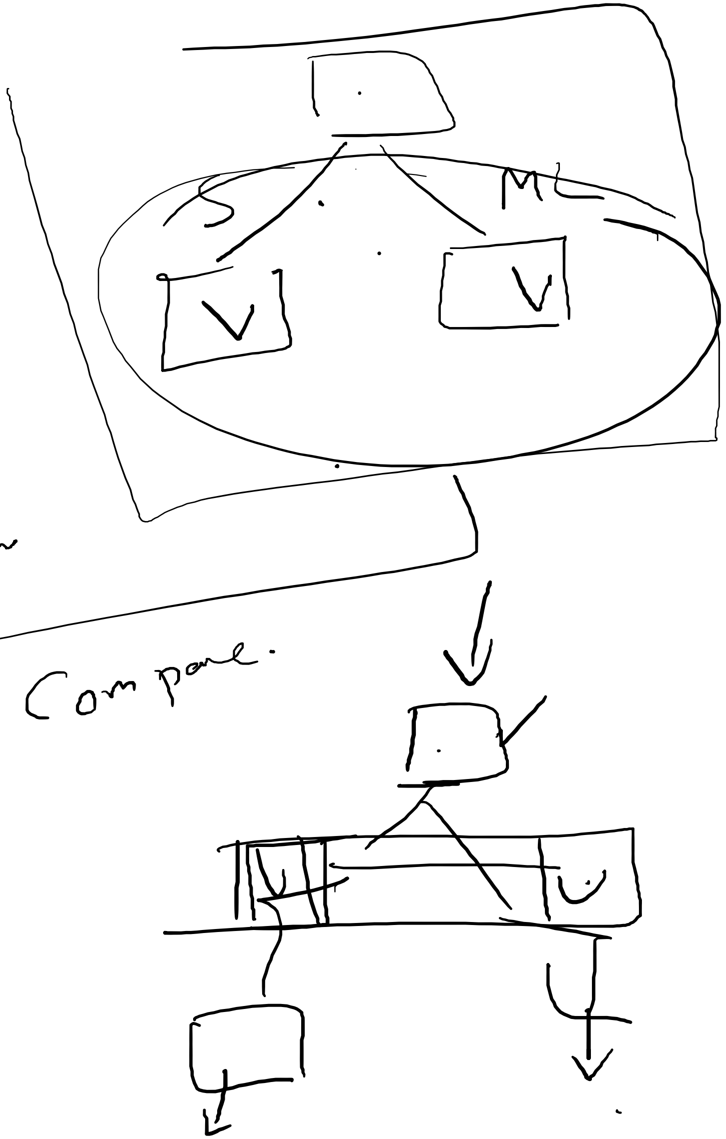
1 2 3

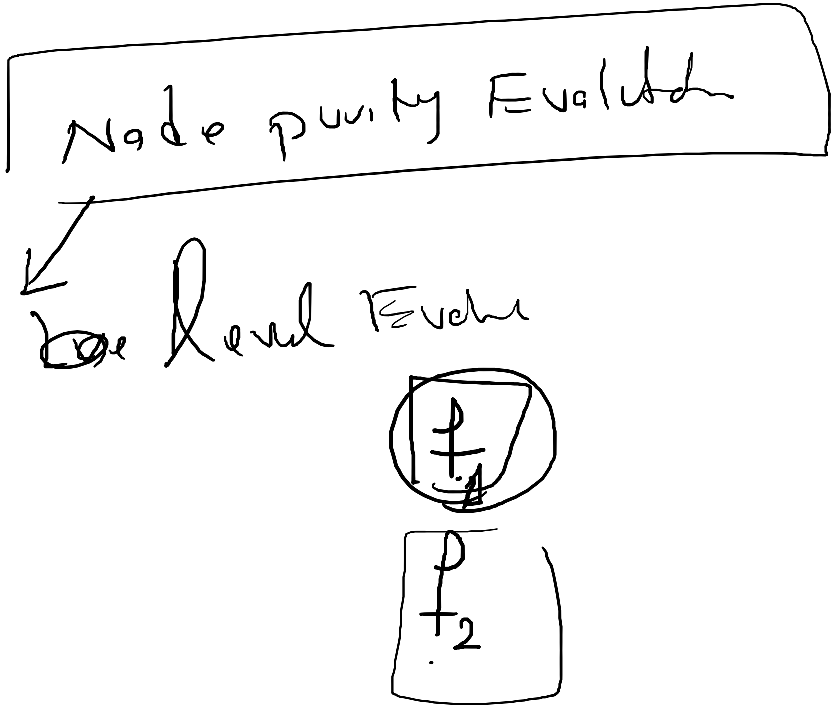


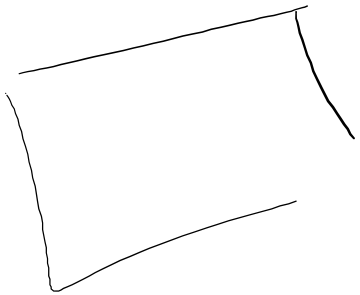

f
k
k-1
k-2
k-3
2
≤ 2^k

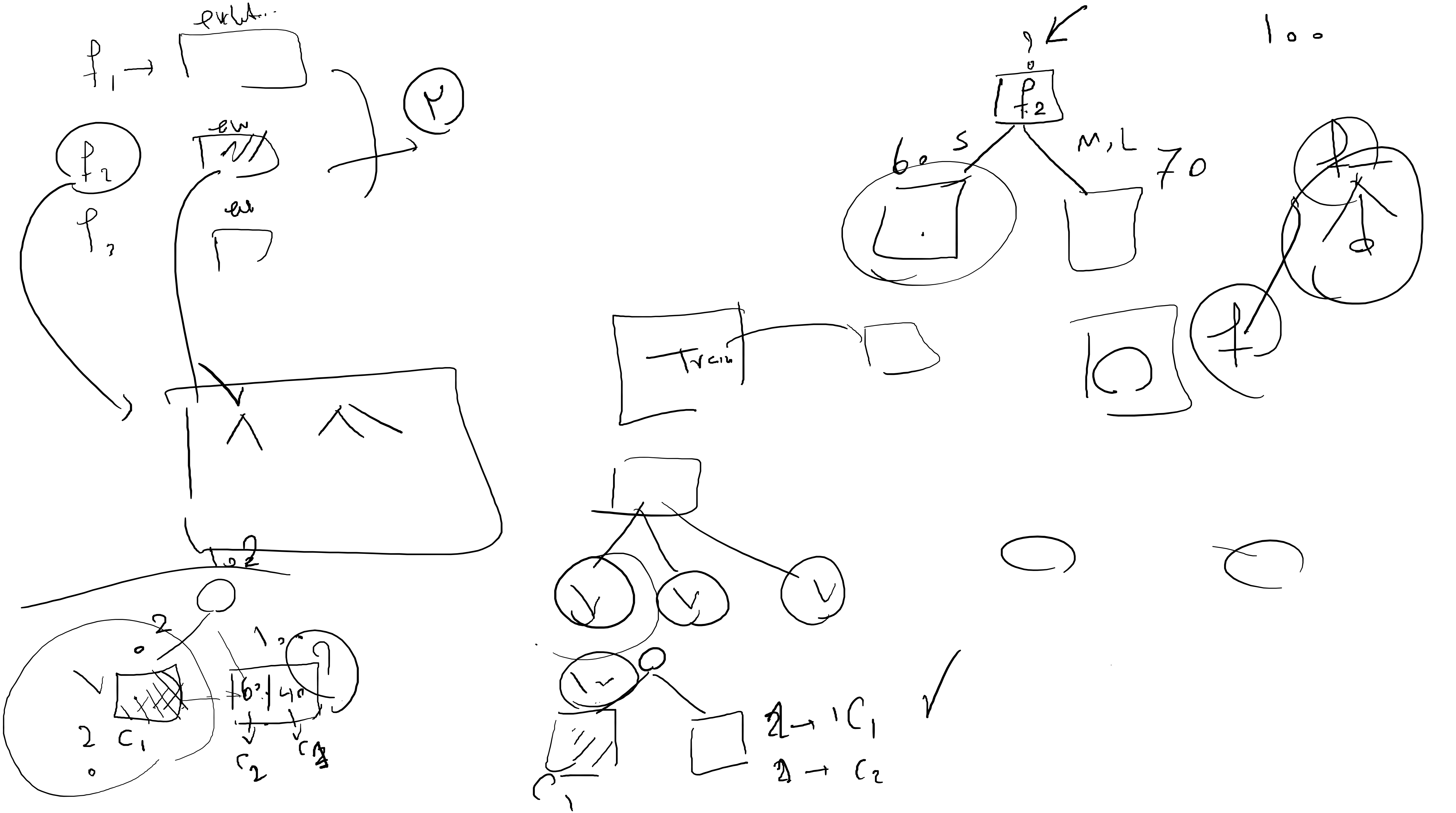

C.45
2
3


V
C1
Ck
100
SM
No
Yes
pure → Node
→ Criterion

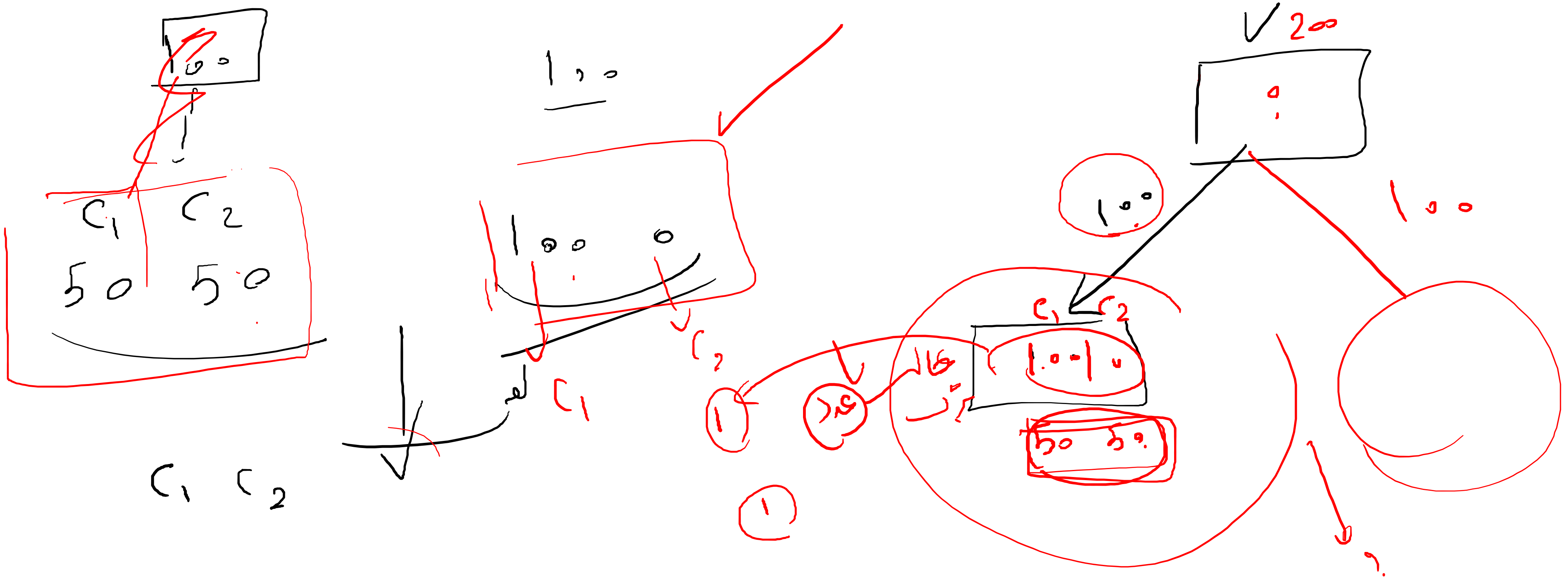

S
ML
V
V
Compare.


Node purity Evalution
Tree Level Evalu
f1
f2





evalt...
f1 →
ew
f2
f3
ew
60 S
M, L
70
100
f2
Train
1, 2
60 | 40
2 C1
C2
C1
2 → C1
2 → C2
C1



$100 = N_t$

$$-\left[\left(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2}\right)\right]$$

$$= -\log\frac{1}{2} = \log 2$$

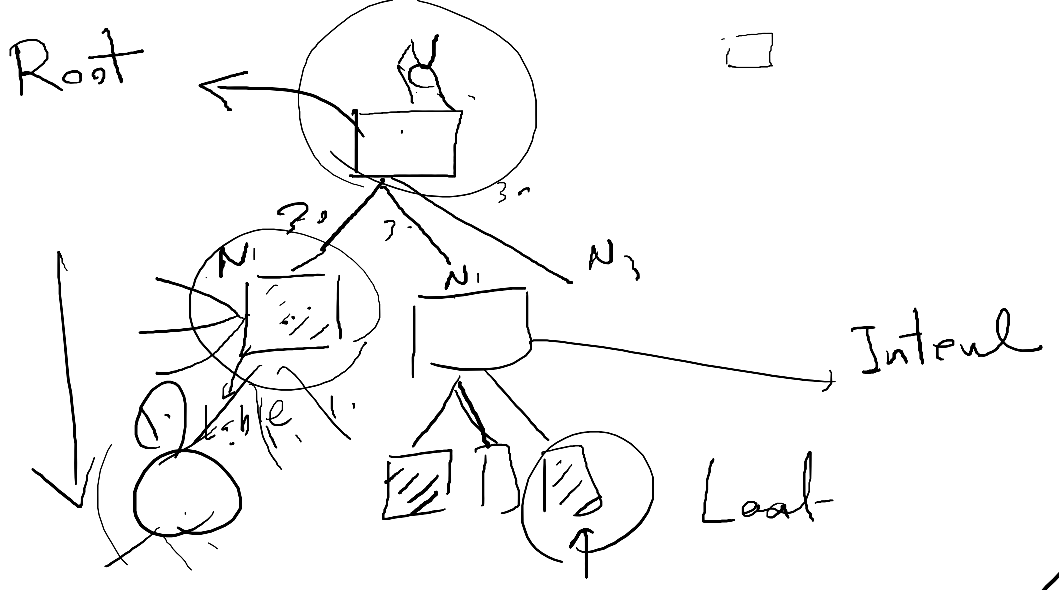
$$\boxed{\log K}$$

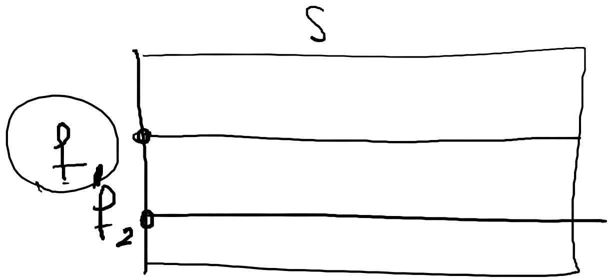
$N_t^1 = 100 \qquad P(1|t) = 1$

$N_t^2 = 0 \qquad P(2|t) = 0$

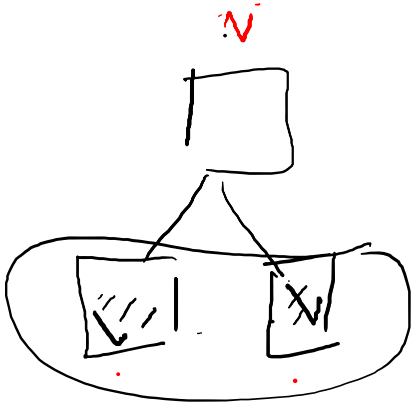
$$-\sum p(i|t)\log p(i|t) = -(0\times\log 0 + 1\times\log 1)$$

$$= 0$$





Evaluation of features

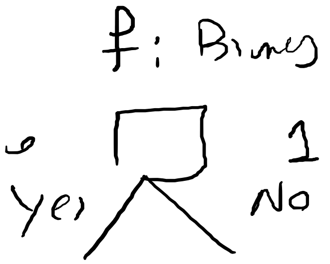


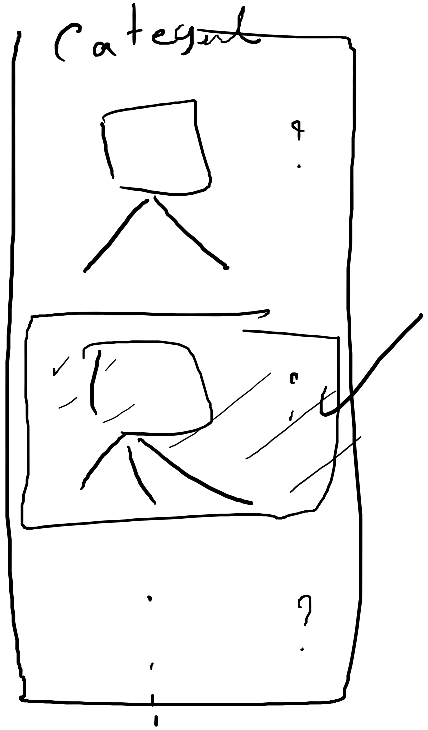
Evaluate Splitting

Evaluate Node

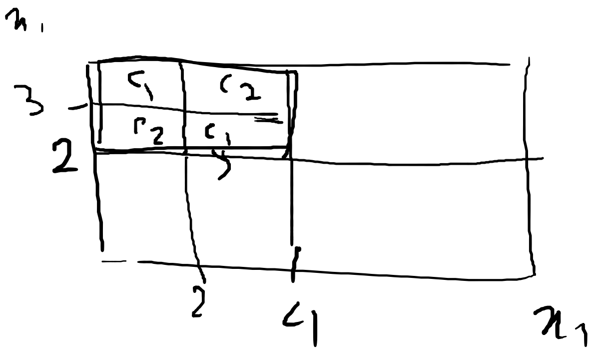
Gini, Entropy & Error e.g.

$I(N_i)$





$$= I(N) - \sum I(N_i) \times \frac{N_i}{N}$$



if $x_1 < 4$

if $x_2 > 2$

if $x_1 > 2$

if $x_2 < 3$

c_1

هستند $p(i|t)$ تعداد عناصر در نود t ام $= N_t$

N_t^i : تعداد عناصر در نود t از کلاس i ام

Measures for Selecting the Best Split

$\exists \ell \quad p(\ell|t) \gg p(i|t) \quad i \neq \ell$

$$p(i|t) = \frac{N_t^i}{N_t}$$

$p(1|t) =$ $\quad C_1 \quad N_t^1$

$p(k|t) \quad C_k \quad N_t^k$

Greedy approaches:

$$\text{Entropy}(t) = \boxed{-\sum_{i=0}^{c-1} p(i|t)\log_2 p(i|t),}$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i [p(i|t)],$$

Let $p(i|t)$ denote the fraction of records belonging to class i at a given node t.

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

$$\boxed{\text{Gain ratio}} = \frac{\Delta_{\text{info}}}{\text{Split Info}}$$

$$p(v_i) = \frac{N_i}{N}$$

min $\boxed{\text{Split Info} = -\sum_{i=1}^{k} P(v_i)\log_2 P(v_i)}$

Node N_1	Count
Class=0	0
Class=1	6

Gini $= 1 - (0/6)^2 - (6/6)^2 = 0$
Entropy $= -(0/6)\log_2(0/6) - (6/6)\log_2(6/6) = 0$
Error $= 1 - \max[0/6, 6/6] = 0$

Node N_2	Count
Class=0	1
Class=1	5

Gini $= 1 - (1/6)^2 - (5/6)^2 = 0.278$
Entropy $= -(1/6)\log_2(1/6) - (5/6)\log_2(5/6) = 0.650$
Error $= 1 - \max[1/6, 5/6] = 0.167$

Node N_3	Count
Class=0	3
Class=1	3

Gini $= 1 - (3/6)^2 - (3/6)^2 = 0.5$
Entropy $= -(3/6)\log_2(3/6) - (3/6)\log_2(3/6) = 1$
Error $= 1 - \max[3/6, 3/6] = 0.5$

Algorithm 4.1 A skeleton decision tree induction algorithm.

`TreeGrowth` (E, F)

if `stopping_cond`$(E,F) = \textit{true}$ **then**
    $leaf$ = `createNode()`.
    $leaf.label$ = `Classify`(E).
    return $leaf$.
else
    $root$ = `createNode()`.
    $root.test\_cond$ = `find_best_split`(E, F).
    let $V = \{v | v$ is a possible outcome of $root.test\_cond\}$.
    for each $v \in V$ **do**
        $E_v = \{e \mid root.test\_cond(e) = v$ and $e \in E\}$.
        $child$ = `TreeGrowth`(E_v, F).
        add $child$ as descendent of $root$ and label the edge $(root \rightarrow child)$ as v.
    end for
end if
return $root$.

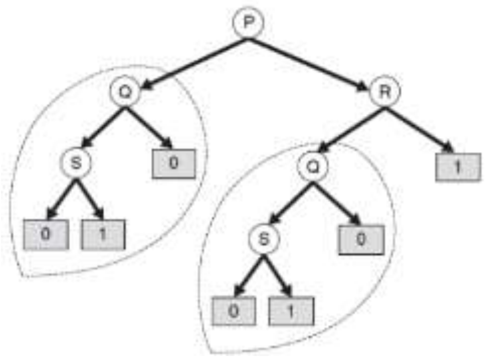
Characteristics of Decision Tree Induction

Decision tree induction is a nonparametric approach

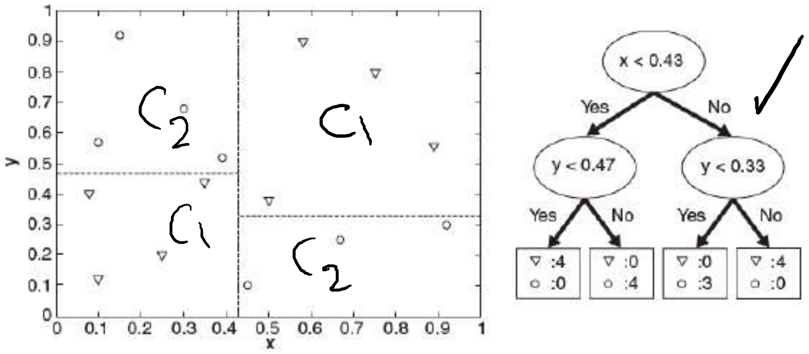
1- Finding an optimal decision tree is an NP-complete problem.

Since most decision tree algorithms employ a top-down, recursive partitioning approach, the number of records becomes smaller as we traverse down the tree. At the leaf nodes, the number of records may be too small to make a statistically significant decision about the class representation of the nodes. This is known as the **data fragmentation** problem. One possible solution is to disallow further splitting when the number of records falls below a certain threshold.

Tree replication problem.



Linear or nonlinear

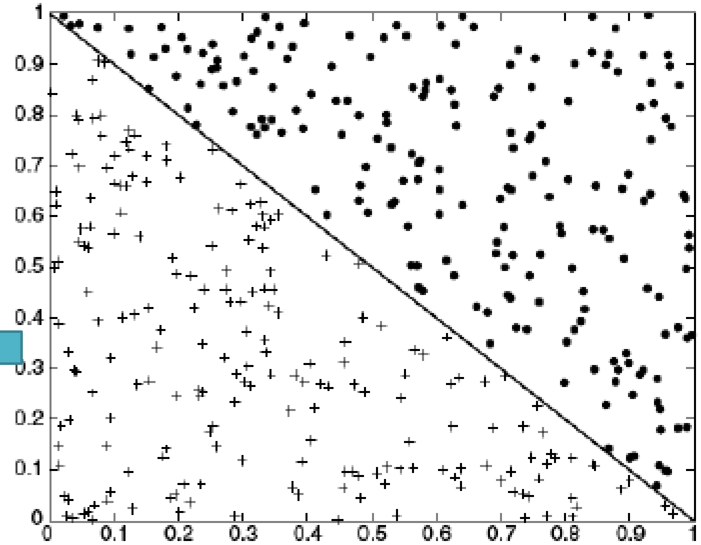


An **oblique decision tree** can be used to overcome this limitation because it allows test conditions that involve more than one attribute. The data set given in Figure 4.21 can be easily represented by an oblique decision tree containing a single node with test condition

$$x + y < 1.$$

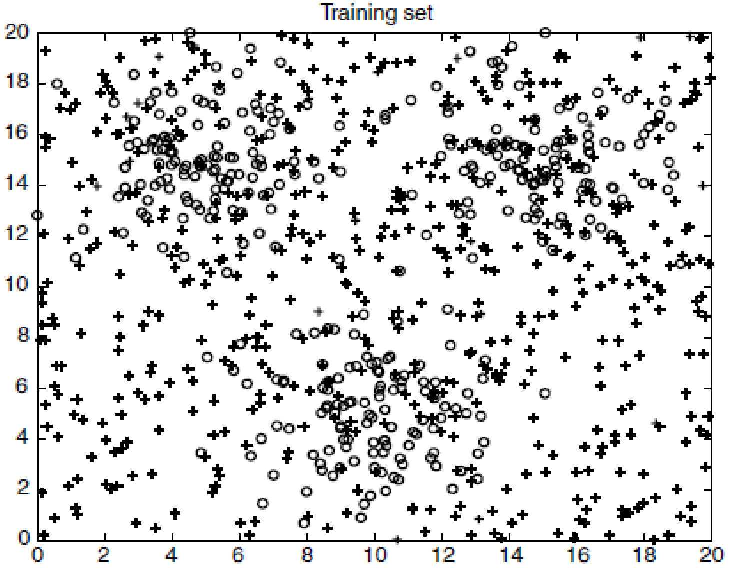
Although such techniques are more expressive and can produce more compact trees, finding the optimal test condition for a given node can be computationally expensive.

Solution



Tan Dal

Overfitting



x_1

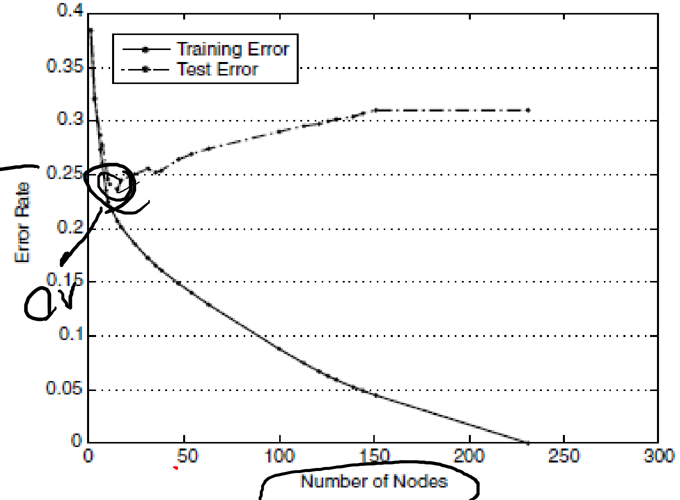
x_2

f_1

f_2

f_2

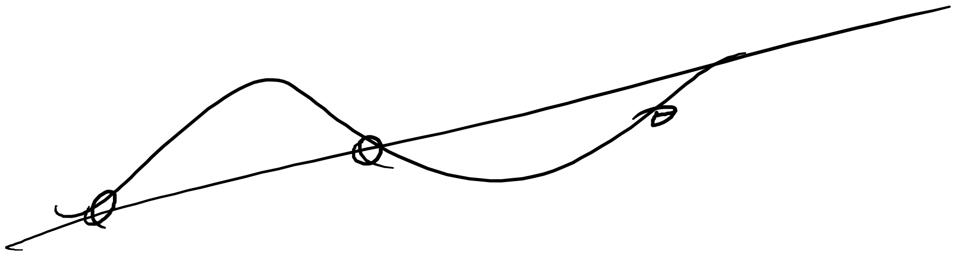
Depth

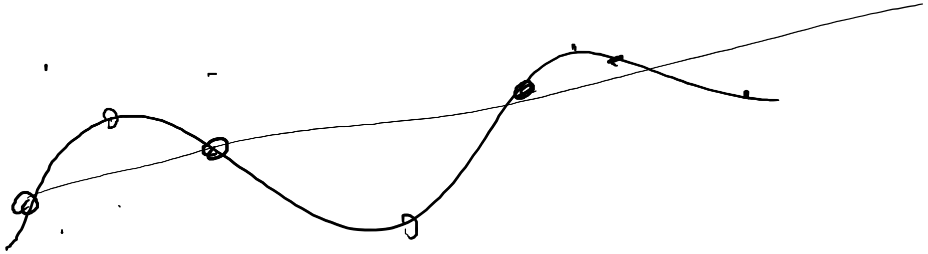


Ov

overfit

- Presence of noise
- Lake of representative samples





Decision Tree

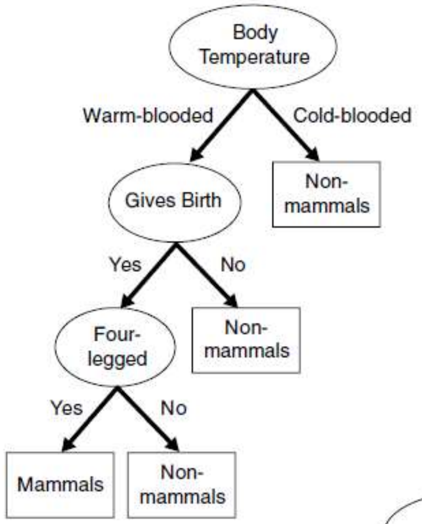
حساس

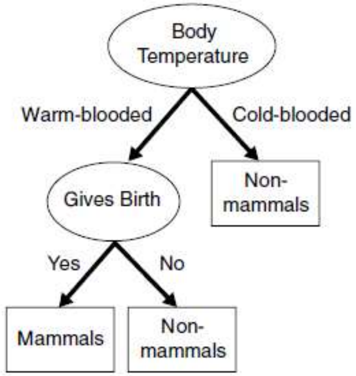
Train

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

Test

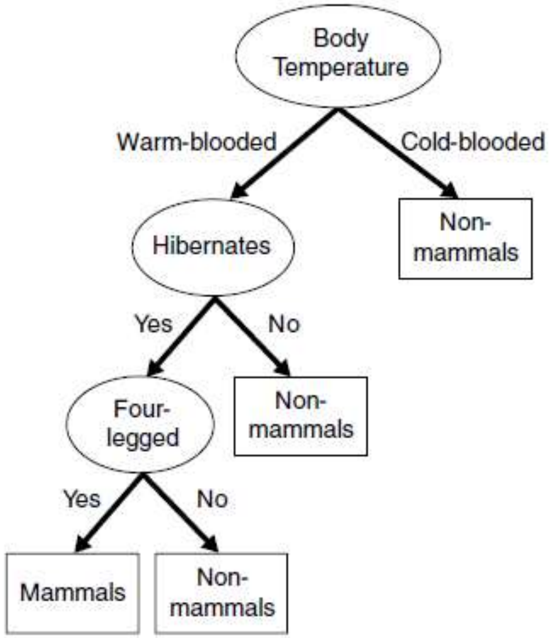
Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
human	warm-blooded	yes	no	no	yes





Train

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
salamander	cold-blooded	no	yes	yes	no
guppy	cold-blooded	yes	no	no	no
eagle	warm-blooded	no	no	no	no
poorwill	warm-blooded	no	no	yes	no
platypus	warm-blooded	no	yes	yes	yes

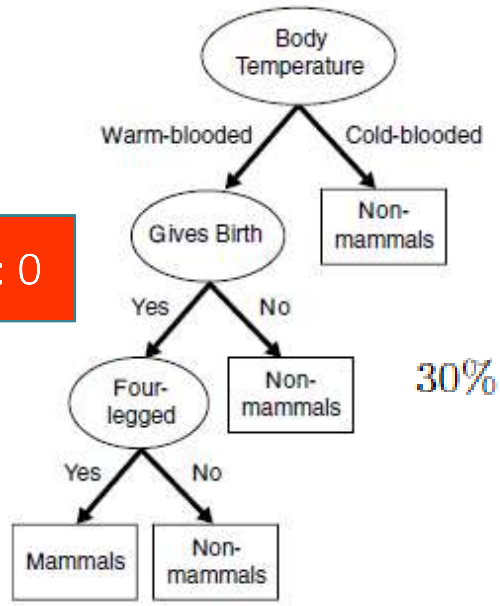


Overfitting Due to Preser

Train

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

Train error: 0

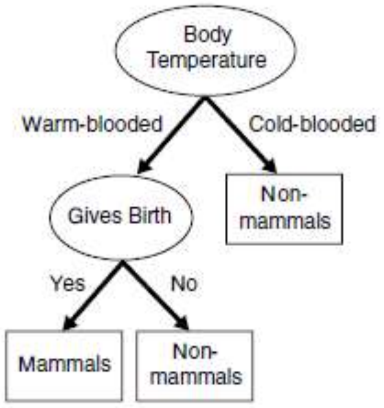


30%

(a) Model M1

Test

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
human	warm-blooded	yes	no	no	yes
pigeon	warm-blooded	no	no	no	no
elephant	warm-blooded	yes	yes	no	yes
leopard shark	cold-blooded	yes	no	no	no
turtle	cold-blooded	no	yes	no	no
penguin	cold-blooded	no	no	no	no
eel	cold-blooded	no	no	no	no
dolphin	warm-blooded	yes	no	no	yes
spiny anteater	warm-blooded	no	yes	yes	yes
gila monster	cold-blooded	no	yes	yes	no

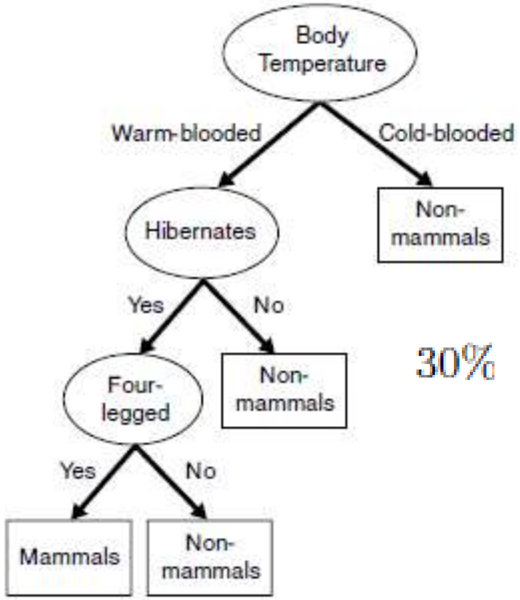


Train error: 20%

error rate (10%)

small number of training records

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
salamander	cold-blooded	no	yes	yes	no
guppy	cold-blooded	yes	no	no	no
eagle	warm-blooded	no	no	no	no
poorwill	warm-blooded	no	no	yes	no
platypus	warm-blooded	no	yes	yes	yes



30%

lin Regression

logistic

SVM

Bayes | Naive Baye

KNN , Perceptron (Variants)

→ Decision Tree

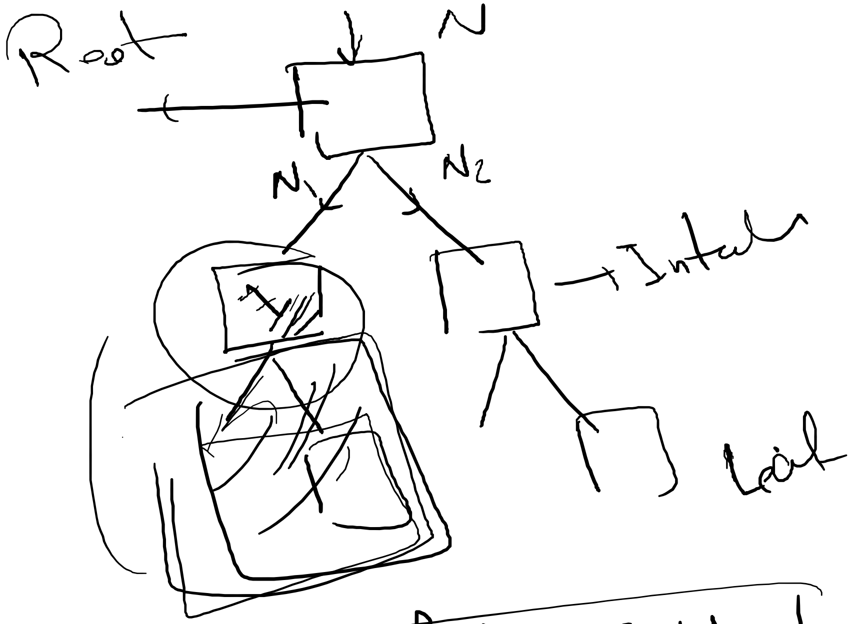
$P(x|C_i)P(C_i)$

→ Evaluation

→ combine : Ensemble method

x_1 x_2

Zaki



Feature Evaluation

Split Evaluation

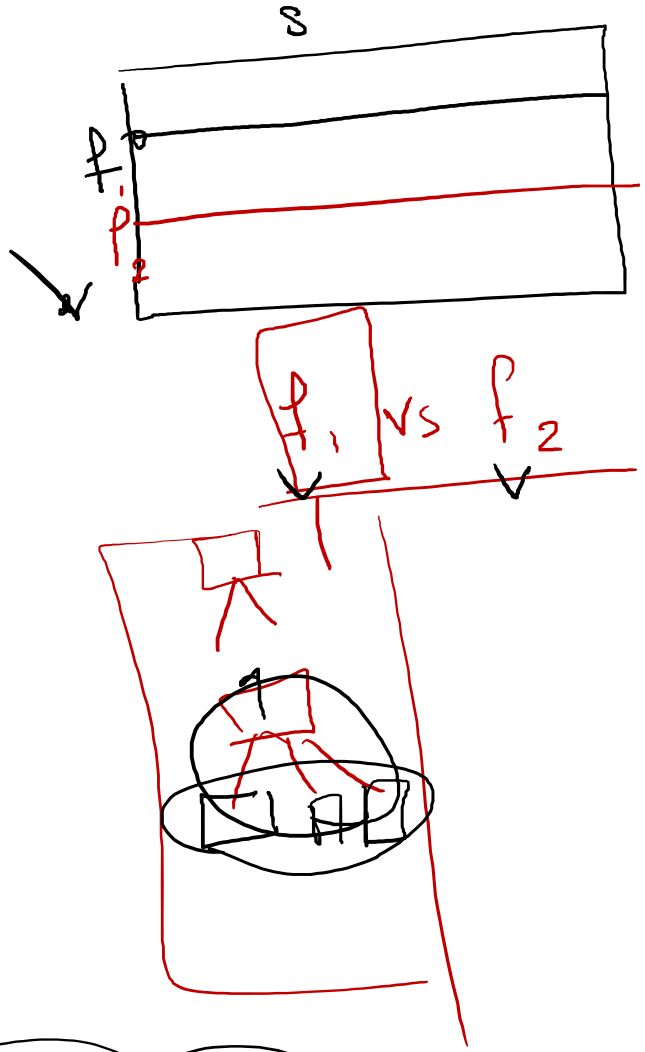
Node Evaluation

Gini

Entropy

Error rate

$$\text{Gain} = I(V) - \sum \frac{N_i}{N} I(V_i)$$



Decision Trees

Prepruning :Early Stopping Rule

a more restrictive stopping condition
stop expanding a leaf node when the observed gain in impurity measure is low

Depth.

Post-pruning

decision tree is initially grown to its maximum size
tree-pruning step
replacing a subtree with a new leaf node

J(V)

J(V)

Evaluating the Performance of a Classifier

Train Test

- accuracy or error rate computed from the test set can used to compare different classifiers
- class labels of test records must be known

model

5-fold

k-fold

80%

Holdout Method

1. labeled examples partitioned into two disjoint sets: training and the test sets
2. classification model is then induced from the training set
3. its performance is evaluated on the test set

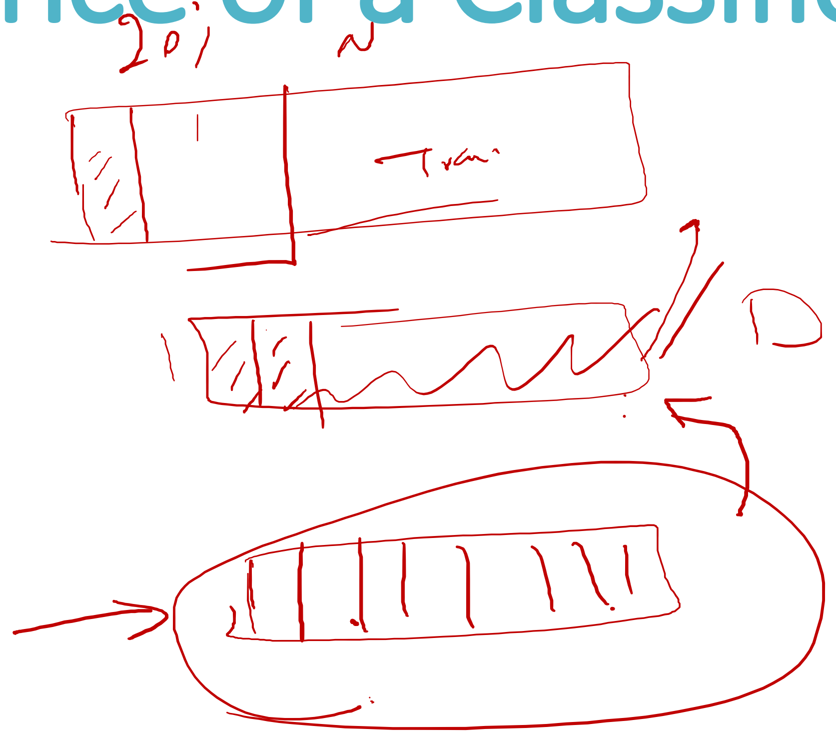
SVM

5

✓ smaller training set size, larger variance of the model

✓ training set is too large, then the estimated accuracy computed from the smaller test set is less reliable

Evaluating the Performance of a Classifier



Random Subsampling

Repeated holdout

Bootstrap

Sampling with replacement

Cross-Validation

each record is used the same number of times for training and exactly once for testing
K-fold Cross-Validation

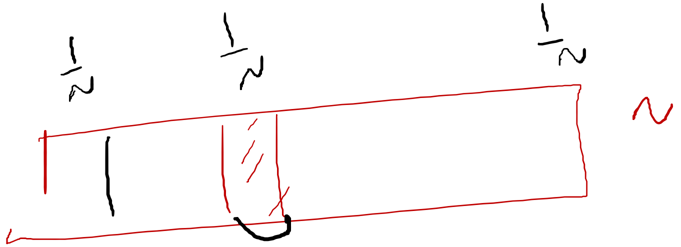
sampling

Cross-Validation

Bootstrap

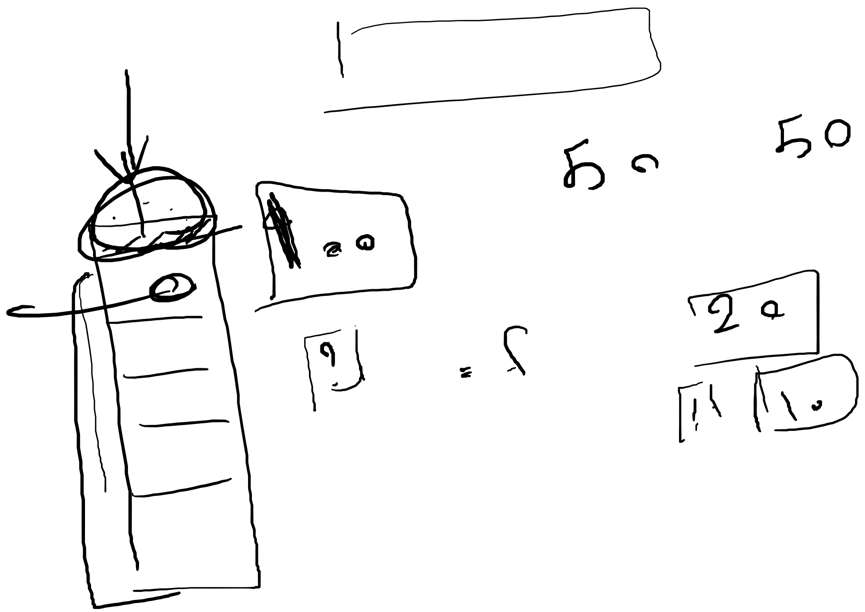
$1 - (1 - 1/N)^N. \Rightarrow 1 - e^{-1} = 0.632.$

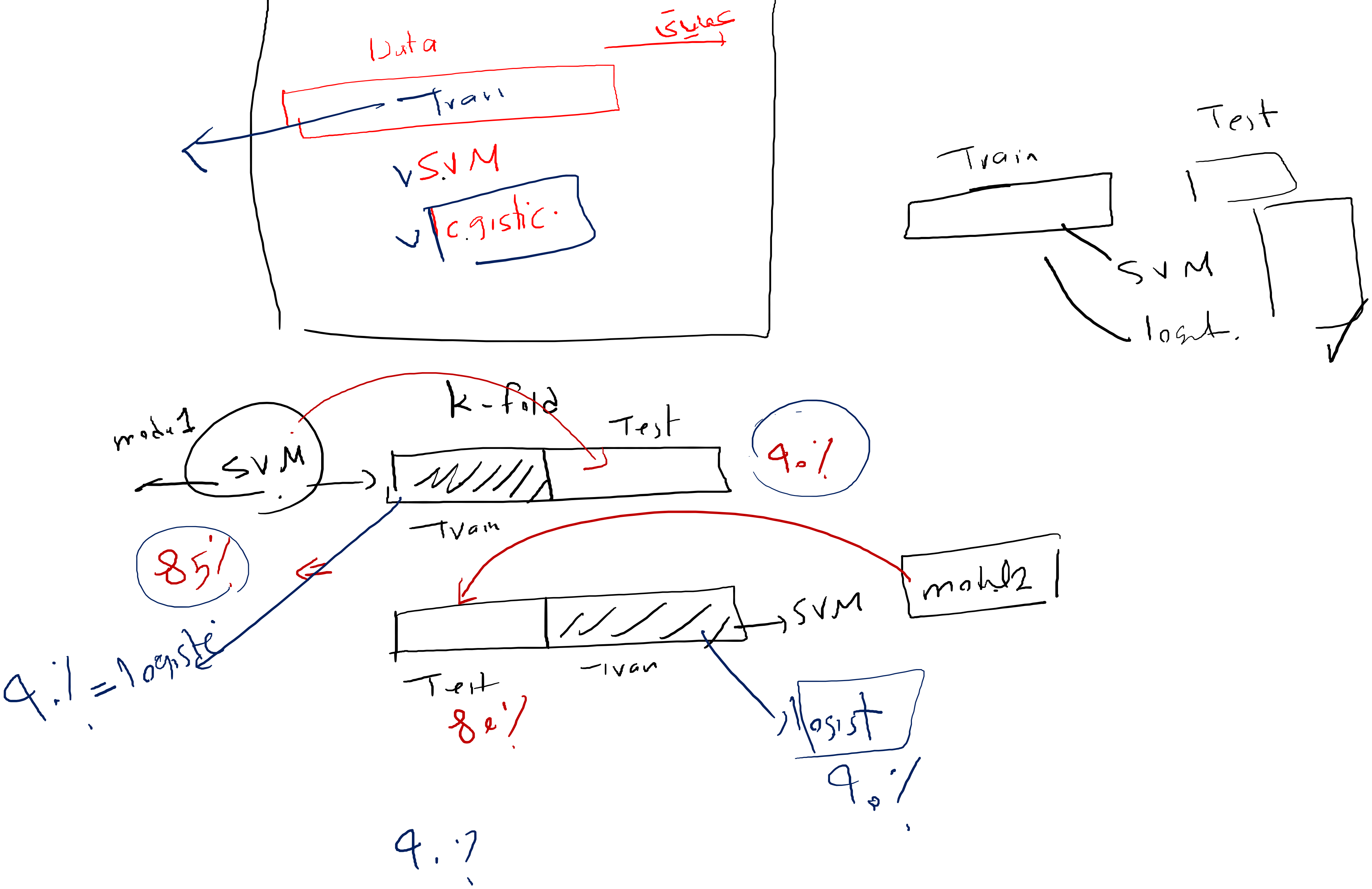
Leave on out



$1 - (1 - \frac{1}{N})^N$

$(1 - \frac{1}{N})(1 - \frac{1}{N}) \cdots (1 - \frac{1}{N}) =$




Data
Train
SVM
logistic
Train
Test
SVM
logit.
k-fold
model
SVM
Test
۹۰٪
85%
Train
model2
SVM
Test
Train
۸۰٪
logist
۹۰٪
۹۰٪ = logistic
۹۰٪

Metrics for class imbalance problem

Imbalance

- ✓ Data sets with imbalanced class distributions
- ✓ in credit card fraud detection, fraudulent transactions are outnumbered by legitimate transactions
- ✓ accuracy measure, used extensively for classifiers, may not be well suited for evaluating models derived from imbalanced data sets

example : 1% of the credit card transactions fraudulent,
a model that predicts every transaction as legitimate
accuracy 99%
it fails to detect any of the fraudulent activities.

binary classification, the rare class is often denoted as the positive class against negative class

		Predicted Class	
		+	−
Actual Class	+	f_{++} (TP)	f_{+-} (FN)
	−	f_{-+} (FP)	f_{--} (TN)

confusion matrix

Imbalance

Precision : fraction of records that actually turns out to be positive in the group the classifier has declared as a positive class

$$\boxed{\text{Precision}, p = \frac{TP}{TP + FP}}$$

Recall measures the fraction of positive examples correctly predicted by the classifier

$$\boxed{\text{Recall}, r = \frac{TP}{TP + FN}}$$

maximizes both precision and recall

Imbalance

Precision and recall can be summarized into another metric known as the F1 measure

$$\boxed{F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}}.}$$

tends to be closer to the smaller of the two numbers
a high value of *F*1-measure ensures that both precision and recall are reasonably high

$$\boxed{\text{Weighted accuracy} = \frac{w_1 TP + w_4 TN}{w_1 TP + w_2 FP + w_3 FN + w_4 TN}.}$$