CLASSIFICATION & DECISION TREES

Agenda

- Evaluating classification results
- Decision Trees
  - How to build trees
    - Entropy
    - Gain Ratio
    - Ginni
- Random Forests & Ensemble Methods

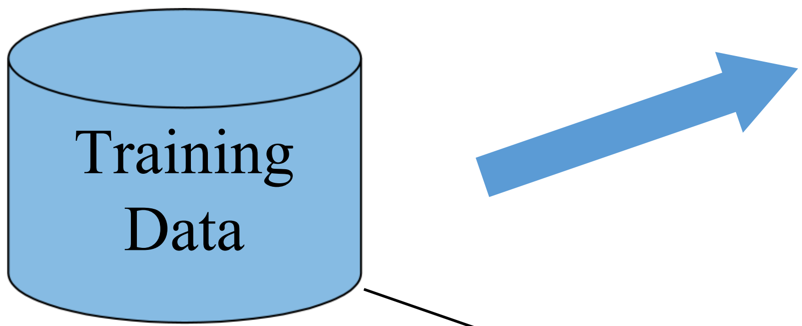
Accuracy Measures (discussed in class)

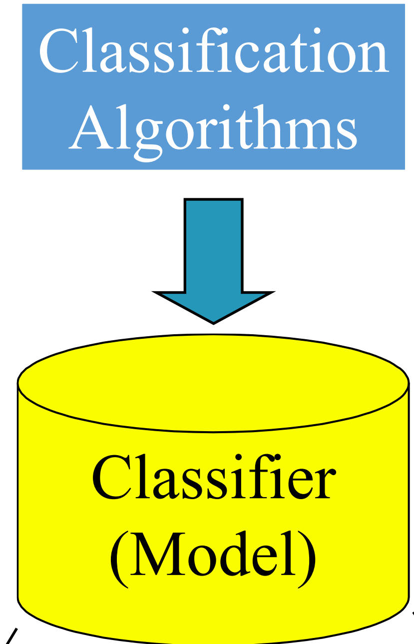
- Accuracy
- Precision
- Recall
- F1 Score
- ROC- AUC Curves
- Recall Precision Trade-off

Try your hands?

Name	Qualification	Exp.	Loan?
Amar	B.Tech	3	no
Bindu	B.Tech	7	yes
Chetan	MBBS	2	yes
Jim	B.Com	7	yes
Dave	B.Tech	6	no
Anne	B.Com	3	no

Model Construction

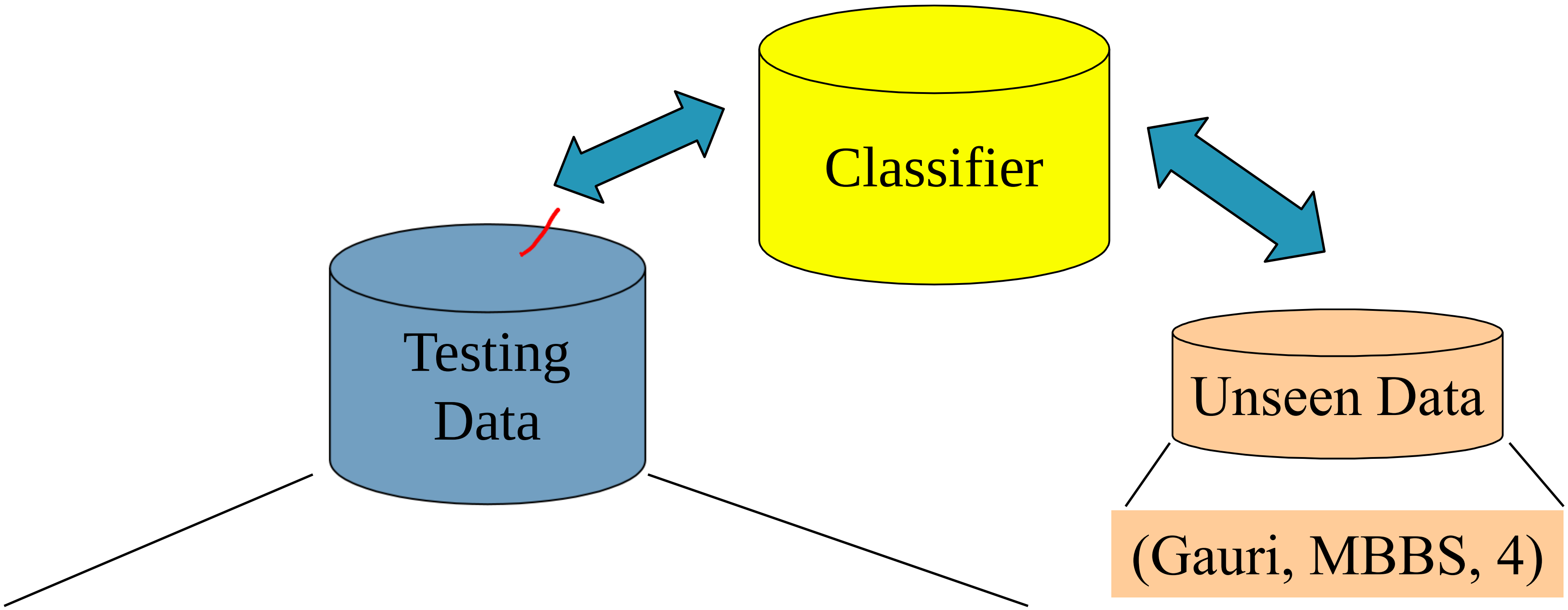




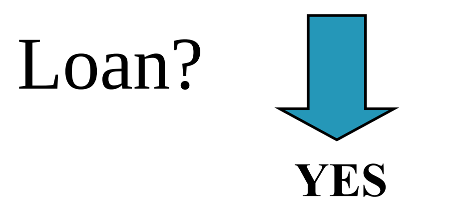
Name	Qualification	Exp.	Loan?
Amar	B.Tech	3	no
Bindu	B.Tech	7	yes
Chetan	MBBS	2	yes
Jim	B.Com	7	yes
Dave	B.Tech	6	no
Anne	B.Com	3	no

IF qualification = ‘MBBS’
OR exp. > 6
THEN loan = ‘yes’

Model Usage



Name	Qualification	Exp.	Loan?
Tarun	B.Tech	2	no
Maya	B.Com	7	no
Hem	MBBS	5	yes
David	B.Tech	7	yes



Decision Tree Induction

- Q: Predict whether an individual will buy a computer?

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Algorithms

- Basic algorithm (a greedy algorithm)
  - 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
  - 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
  - There are no samples left

Information Gain (ID3)

- Select the attribute with the highest information gain
- Let pi be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, 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 needed (after using A to split D into v partitions) to classify D:

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

- Information gained by branching on attribute A

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

Attribute Selection

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

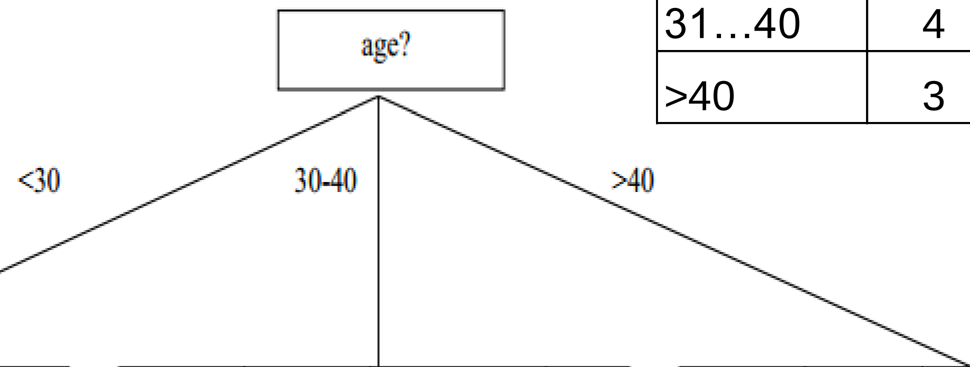
- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

- Calculate information needed to classify:

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

Split on Age?

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971



income	student	credit_rating	Class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	Class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

income	student	credit_rating	Class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Split on Age?

$$Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

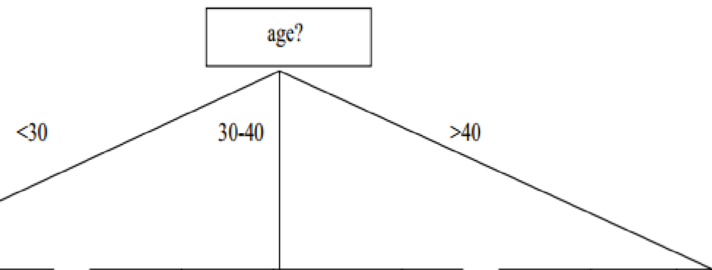
$$Gain(credit\_rating) = 0.048$$

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's.

Hence,

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Continue Process



income	student	credit_rating	Class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	Class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

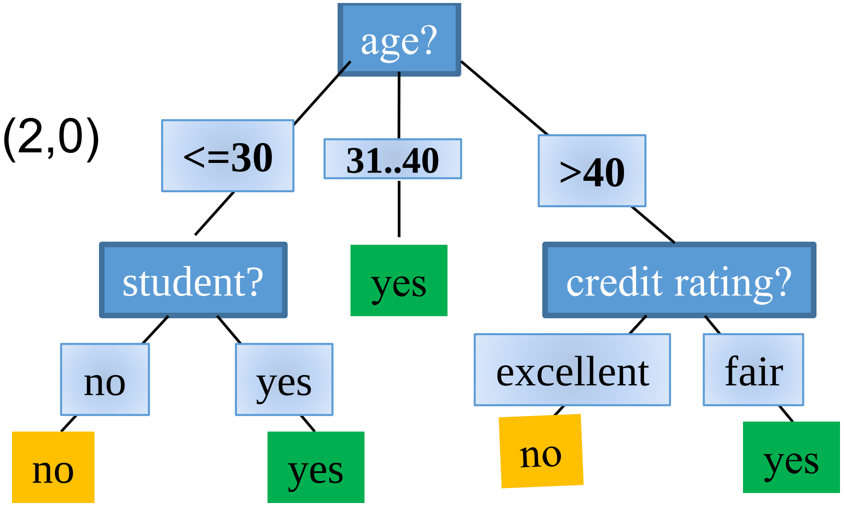
income	student	credit_rating	Class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

Info(D’) = I(2,3)

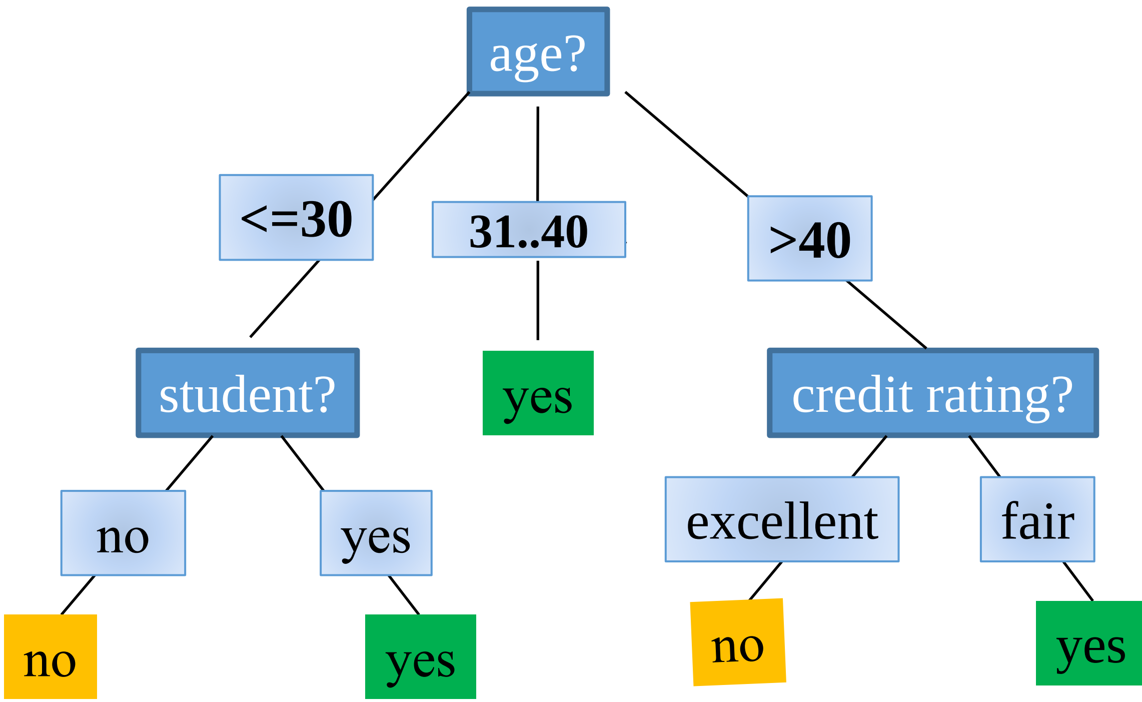
Info_student(D’) = 3/5 I(0,3) + 2/5 I(2,0)

Gain(student) = ?

….



Decision Tree Construction



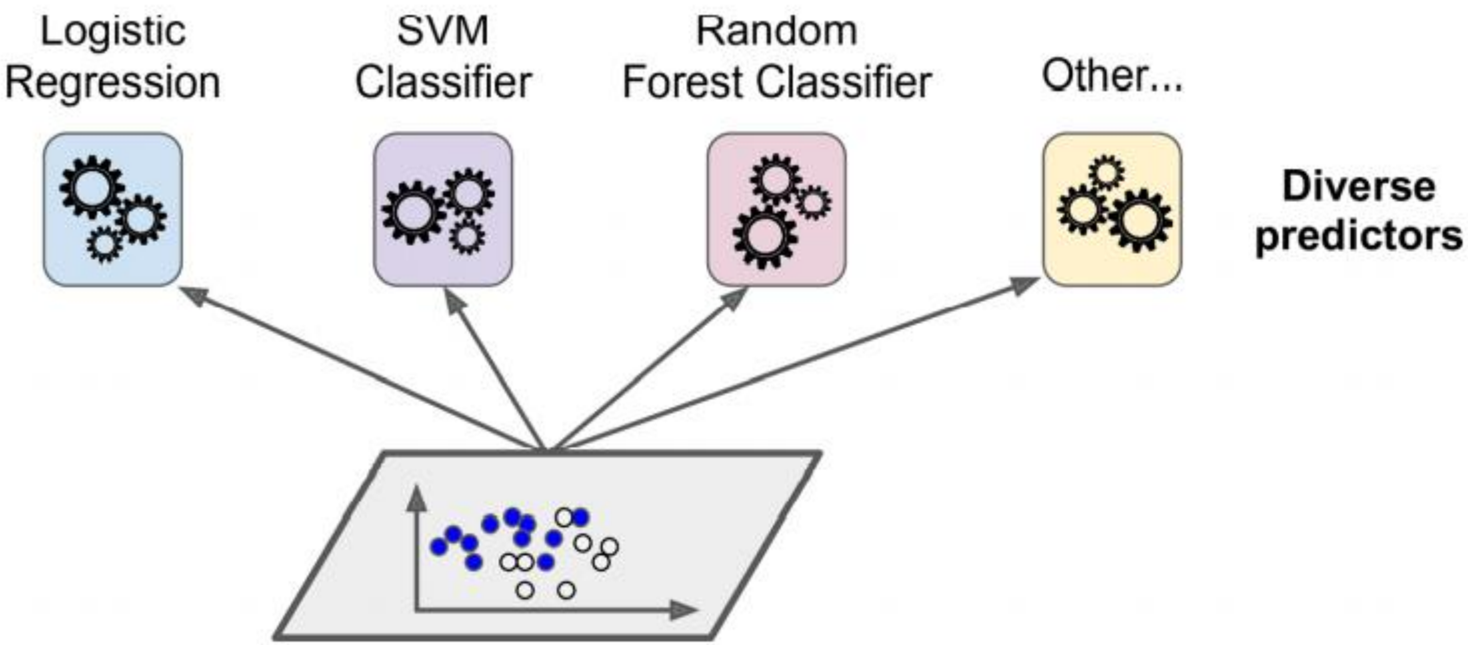
age	income	student	credit_rating	buys
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Over-fitting and Pruning

- **Over-fitting:** A tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid over-fitting
  - **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

Ensemble Methods & Random Forests

- Voting Classifiers



Random Forest

- Advantages of Bagging
  - Easy to implement
  - Reduces variance
  - More unbiased estimate of the test error

- Random Forest Algorithm
  - Same *m datasets D1, D2,….Dm from D with replacement*
  - *For each Di train a full classifier*
    - *before each split randomly subsample k≤d features*
  - *The final classifier is the average of m classifiers*
  - *There are 2 hyper-parameters m,k… a good approximation for k is SQRT(d)—where d is the number of features*
  - *OOB Evaluation*

Ada Boost

Input: ℓ, α, $\{(\mathbf{x}_i, y_i)\}$, $\mathbb{A}$

$H_0 = 0$

$\forall i : w_i = \frac{1}{n}$

for *t=0:T-1* **do**

  $h = \operatorname{argmin}_h \sum_{i:h(\mathbf{x}_i)\neq y_i} w_i \qquad [h = \mathbb{A}\left((w_1, \mathbf{x}_1, y_1), ..., (w_n, \mathbf{x}_n, y_n)\right)]$

  $\epsilon = \sum_{i:h(\mathbf{x}_i)\neq y_i} w_i$

  if $\epsilon < \frac{1}{2}$ **then**

    $\alpha = \frac{1}{2}\ln(\frac{1-\epsilon}{\epsilon})$

    $H_{t+1} = H_t + \alpha h$

    $\forall i : \quad w_i \leftarrow \frac{w_i e^{-\alpha h(\mathbf{x}_i) y_i}}{2\sqrt{\epsilon(1-\epsilon)}}$

  else

    return (H_t)

  end

end

return (H_T)

Gradient Boosting

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$.
2. For $m = 1$ to M:
   (a) For $i = 1, 2, \ldots, N$ compute

   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$

   (b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm},\ j = 1, 2, \ldots, J_m$.
   (c) For $j = 1, 2, \ldots, J_m$ compute

   $$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$

   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.
3. Output $\hat{f}(x) = f_M(x)$.

Source: The Elements Of Statistical Learning, Data Mining Inference And Prediction, 2^{ND} Ed, Pg. 640

THANK YOU! ☺