Parul® University

Data Mining and Warehousing

Prof. Prashant Sahatiya, Assistant Professor
Information Technology

CHAPTER-6

Classification

Classification

- Approach used by programs to use data for learning from it and predict new observations or classification.
- Predict results in discrete output.
- E.g. Cancer type detection, Spam/non spam mail detection.

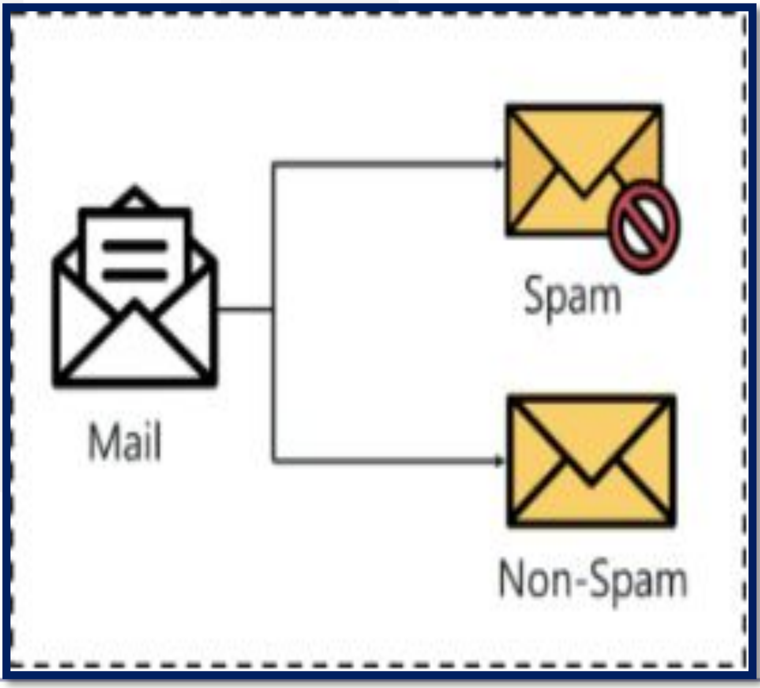


Image source : Google

Prediction

- Predicts continuous valued output using some parameters of the given data.

- E.g. Stock market prediction, House price prediction, Rainfall prediction.

Supervised learning

- It's a process of making a program learn to map an input to a particular output.
- Learning happens using labelled data.
- For correct output the program learned successfully.
- Used to predict the value of unknown data that may arise in future.
- E.g. Teacher-student scenario.

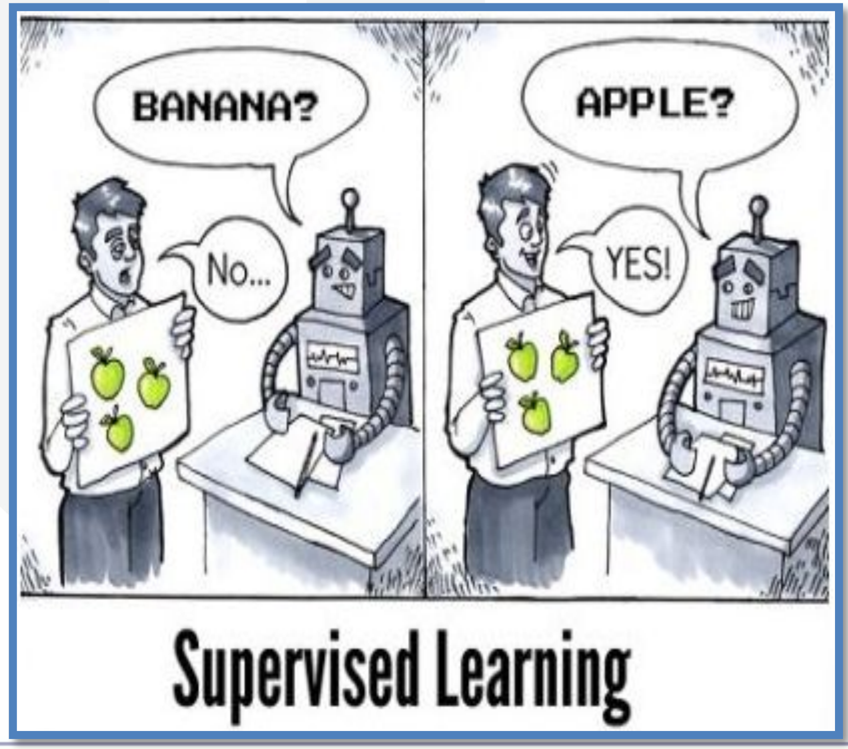


Image source : Google

Approach to Classification

- Two Step Approach

Learn Model

- Collection of rows, each with some attribute one of which being class is referred to as training data.
- Find a function that takes input as attributes and predict class.

Apply Model

- The records whose class label is not know is referred to as testing data.
- Use the model created to classify testing records.

Decision Tree Induction

- What is decision tree?
  - Tree structured model of decisions.
  - Used for predicting best choice mathematically.
  - Starts with single node and branch out to possible outcomes.
  - Each outcomes generates other nodes with other possibilities.
  - Giving it a tree like shape.
  - Internal node – test attribute.
  - Branch corresponding – attribute value.
  - Leaf node – assigns classification

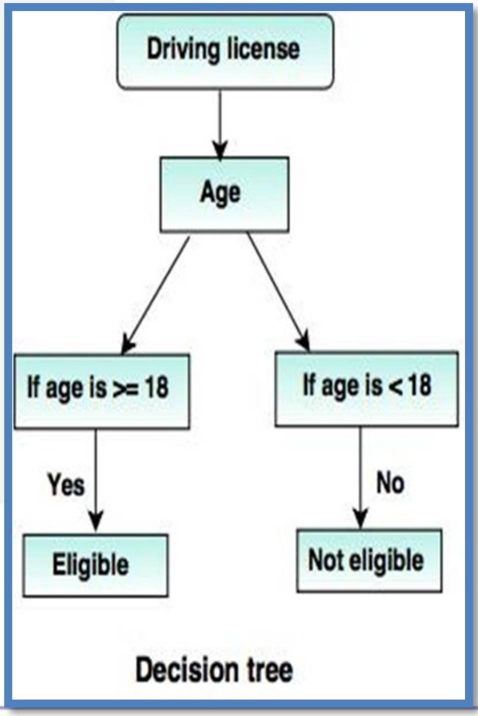


Image source : Google

Decision Tree Creation(The Greedy Approach)

- Makes optimal local choice at each node.
- Reaching approximate global optimal solution.
- For each node take best feature as test condition.
- Splitting node into possible outcomes.
- Repeat till test condition results into leaf node.
- Factors used to identify starting condition are
  - Entropy
  - Information Gain
  - Gini Index

Attribute Selection Measures

- Entropy
  - It's a measure of uncertainty, purity and information content.
    - Consider a sample of training example S
    - P1 is the portion of positive examples in S
    - P2 is the portion of negative example in S
    - Entropy(S) = p1(-log2p1)+p2(-log2p2) = -p1(log2p1) - p2(log2p2)
- Information Gain
  - When a node is split the increase/decrease in the value of entropy is referred to as Information gain.
- For splitting an attribute with highest information gain is selected.

Example – ID3

Age	Income	Student	Credit_Rating	Class : 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

Solution – ID3

- **Class P** : buys_computer = "Yes" (9 records)
- **Class N** : buys_computer = "No" (5 records)
- Total number of Records **14**.
- Now, Information Gain = I (p,n)

$$I(p,n) = -\frac{p}{p+n}log_2\frac{p}{p+n} - \frac{n}{p+n}log_2\frac{n}{p+n}$$

$$I(9,5) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14}$$

$$\boxed{I(9,5) = 0.940}$$

Solution – ID3 (Age <=30,31..40,>40)

Age	Income	Student	Credit_Rating	Buys_Computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes

Age	Income	Stu.	Cr_Rating	Buys
31..40	High	No	Fair	Yes
31..40	Low	Yes	Excellent	Yes
31..40	Medium	No	Excellent	Yes
31..40	High	Yes	Fair	Yes

Age	Income	Stu.	Cr_Rating	Buys
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	No	Excellent	No
>40	Medium	Yes	Fair	Yes
>40	Medium	No	Excellent	No

Solution – ID3 (Age <=30)

- Compute the information gain & Entropy For Age **<=30**,
  - P_i = Yes class = 2
  - N_i = No class = 3

  So, Information Gain = I (p,n)

$$I(p,n) = -\frac{p}{p+n}log_2\frac{p}{p+n} - \frac{n}{p+n}log_2\frac{n}{p+n}$$

$$I(2,3) = -\frac{2}{5}log_2\frac{2}{5} - \frac{3}{5}log_2\frac{3}{5}$$

$$\boxed{I(2,3) = 0.971}$$

Solution – ID3

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

- So the expected information needed to classify a given sample if the samples are partitioned according to age is,
- Calculate entropy using the values from the Table and the formula given below:

$$E(A) = \sum_{i=1}^{v} \frac{P_i + N_i}{p+n} I(P_i, N_i)$$

$$E(Age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2)$$

$$\boxed{E(Age) = 0.694}$$

Solution – ID3

Gain (Age) = I (p, n) – E (Age)

= 0.940 – 0.694

= 0.246

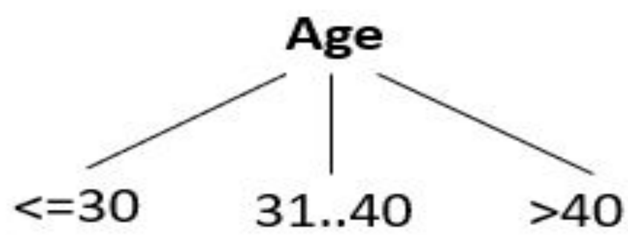
Similarly,

Gain	value
Gain (age)	**0.246**
Gain (income)	0.029
Gain (student)	0.151
Gain (credit_rating)	0.048

So, here we start decision tree with root node **Age**.

Solution – ID3

- Now the age has highest information gain among all the attributes, so select age as test attribute and create the node as age and show all possible values of age for further splitting.



Solution – ID3 (Age <=30)

Age	Income	Student	Credit_Rating	Buys_Computer
<=30	High	No	Fair	No
<=30	High	No	Excellent	No
<=30	Medium	No	Fair	No
<=30	Low	Yes	Fair	Yes
<=30	Medium	Yes	Excellent	Yes

Solution – ID3 (Age <=30)

- Compute Information gain & Entropy for Age with sample $S_{<=30.}$
- For age <=30,
  - Pi = Yes = 2
  - Ni = No = 3

$$I(p, n) = -\frac{p}{p+n} log_2 \frac{p}{p+n} - \frac{n}{p+n} log_2 \frac{n}{p+n}$$

$$I(2,3) = -\frac{2}{5} log_2 \frac{2}{5} - \frac{3}{5} log_2 \frac{3}{5}$$

$$\boxed{I(3,2) = 0.971}$$

Solution – ID3 (Age <= 30, Income)

Income	P_i	N_i	$I(P_i, N_i)$
High	0	2	0
Medium	1	1	1
Low	1	0	0

In above table high (0,2) homogeneous so I(0,2) = 0, Medium equal portion so I(1,1) = 1 & Low I(1,0) = 0.

$$E(A) = \sum_{i=1}^{v} \frac{P_i+N_i}{p+n} I(P_i, N_i)$$

$$E(\text{Income}) = \frac{2}{5} I(0,2) + \frac{2}{5} I(1,1) + \frac{1}{5} I(1,0)$$

$$\boxed{E(\text{Income}) = 0.4}$$

$$\begin{aligned} \text{Gain}(S_{<=30}, \text{Income}) &= I(p, n) - E(\text{Income}) \\ &= 0.971 - 0.4 \\ &= 0.571 \end{aligned}$$

Solution – ID3 (Age <= 30, Student)

student	P_i	N_i	$I(P_i, N_i)$
No	0	3	0
Yes	2	0	0

In above table I (0,3) = 0 & I (2,0) = 0 So E(Student) is 0.

$$\boxed{E(Student) = 0}$$

$$\begin{aligned} Gain(S_{<=30}, Student) &= I(p,n) - E(Student) \\ &= 0.971 - 0 \\ &= 0.971 \end{aligned}$$

Solution – ID3 (Age <= 30, credit_rating)

credit_rating	P_i	N_i	$I(P_i, N_i)$
Fair	1	2	0.918
Excellent	1	1	1

$$E(A) = \sum_{i=1}^{v} \frac{P_i + N_i}{p+n} I(P_i, N_i)$$

$$E(\text{credit\_rating}) = \frac{3}{5} I(1,2) + \frac{2}{5} I(1,1)$$

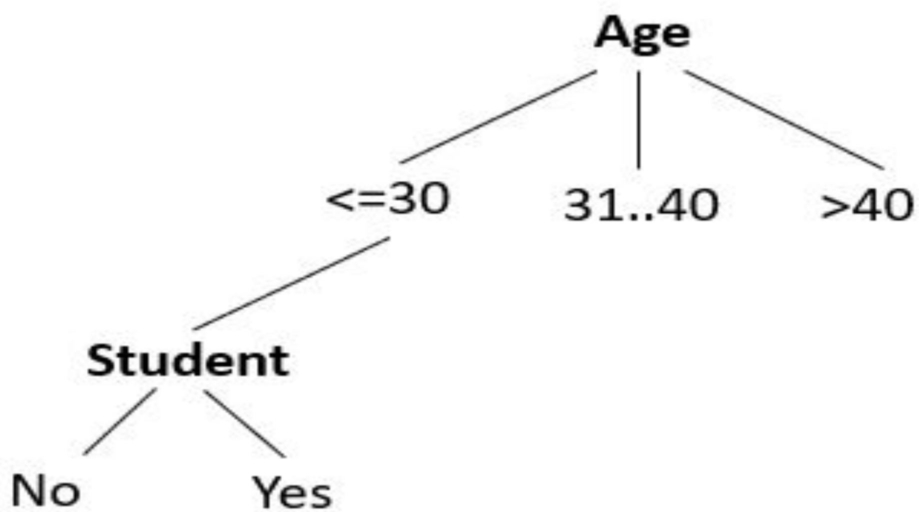
$$\boxed{E(\text{credit\_rating}) = 0.951}$$

$$\begin{aligned} \text{Gain}(S_{<=30}, \text{credit\_rating}) &= I(p, n) - E(\text{credit\_rating}) \\ &= 0.971 - 0.951 \\ &= 0.020 \end{aligned}$$

Solution – ID3 (Age <=30)

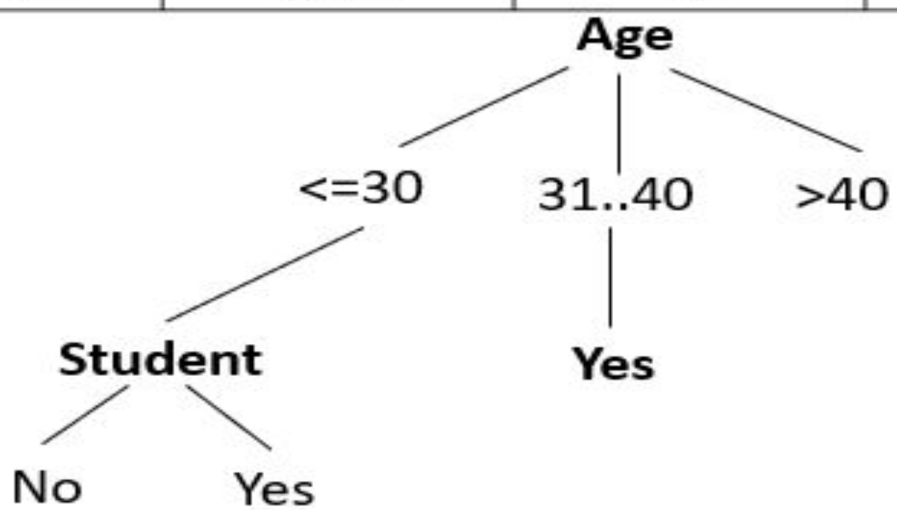
Gain (Age <= 30)	value
Income	0.571
Student	**0.971**
Credit_rating	0.020

As shown in table we get maximum gain for student so, select **student** as leaf node for age <=30



Solution – ID3 (Age 31..40)

Age	Income	Student	Credit_Rating	Buys_Computer
31..40	High	No	Fair	**Yes**
31..40	Low	Yes	Excellent	**Yes**
31..40	Medium	No	Excellent	**Yes**
31..40	High	Yes	Fair	**Yes**



Solution – ID3 (Age > 40)

Age	Income	Student	Credit_Rating	Buys_Computer
>40	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	No	Excellent	No
>40	Medium	Yes	Fair	Yes
>40	Medium	No	Excellent	No

Solution – ID3 (Age > 40)

- Compute Information gain for Age with sample $S_{>40}$.
- For age > 40,
  - Pi = Yes = 3
  - Ni = No = 2

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$I(3,2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$\boxed{I(3,2) = 0.971}$$

Solution – ID3 (Age > 40, Income)

Income	P_i	N_i	$I(P_i, N_i)$
High	0	0	0
Medium	2	1	0.918
Low	1	1	1

$$E(A) = \sum_{i=1}^{v} \frac{P_i+N_i}{p+n} I(P_i, N_i)$$

$$E(Income) = \frac{0}{5} I(0,0) + \frac{3}{5} I(2,1) + \frac{2}{5} I(1,1)$$

$$\boxed{E(Income) = 0.951}$$

Gain ($S_{>40}$,Income)

$= I(p, n) – E(Income)$

$= 0.971 – 0.951$

$= 0.020$

Solution – ID3 (Age > 40, credit_rating)

Credit_rating	P_i	N_i	$I(P_i, N_i)$
Fair	3	0	0
Excellent	0	2	0

$$E(A) = \sum_{i=1}^{v} \frac{P_i+N_i}{p+n} I(P_i, N_i)$$

$$E(\text{credit\_rating}) = \frac{3}{5} I(3,0) + \frac{2}{5} I(0,2)$$

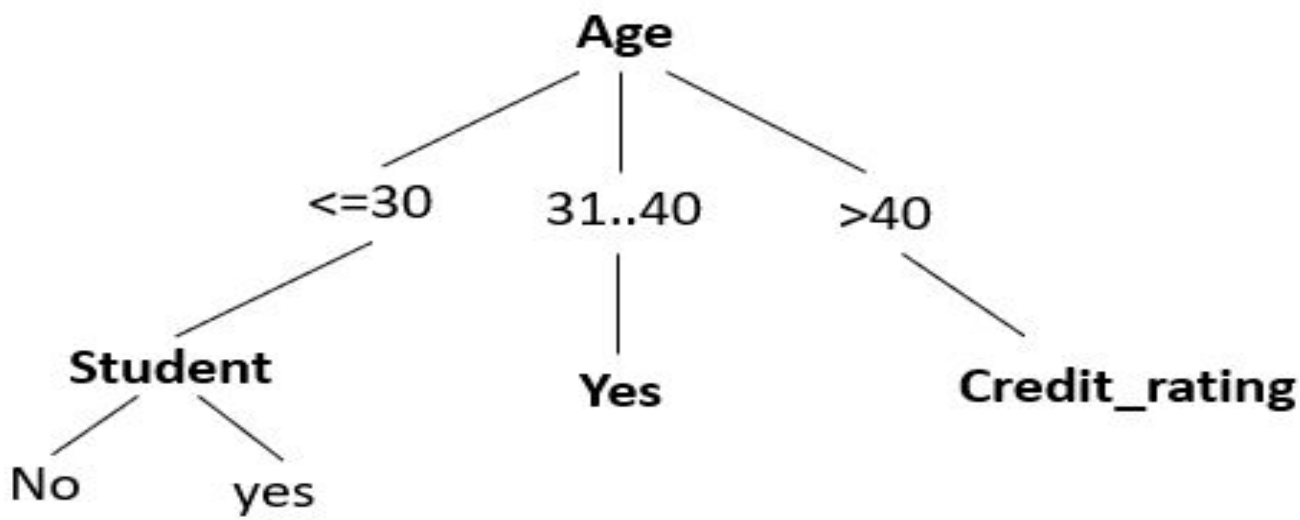
$$\boxed{E(\text{credit\_rating}) = 0}$$

$$\text{Gain}(S_{>40}, \text{credit\_rating}) = I(p, n) - E(\text{credit\_rating})$$
$$= 0.971 - 0$$
$$= 0.971$$

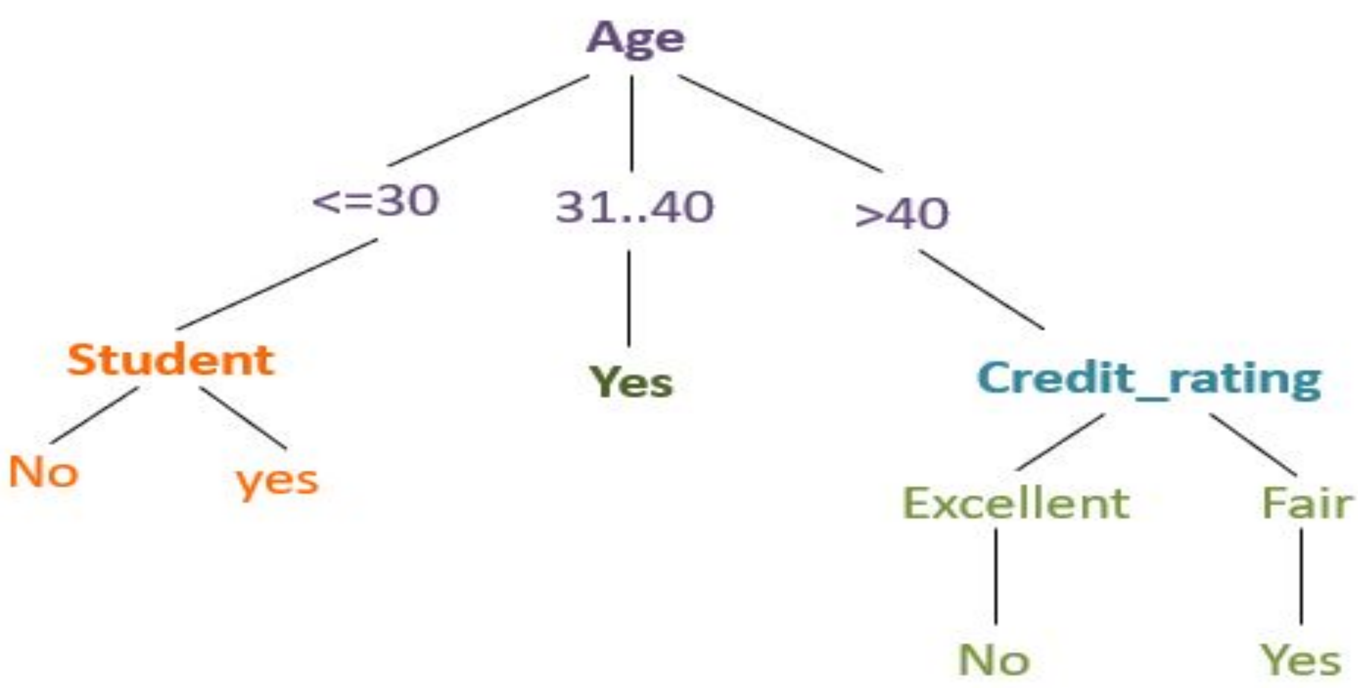
Solution – ID3 (Age > 40)

Gain (Age > 40)	value
Income	0.020
Credit_rating	**0.971**

As shown in table we get maximum gain for credit_rating so, select credit_rating as leaf node for age > 40



Decision Tree – ID3



Classification rules from decision tree

- IF age = “<=30” AND student = “no” THEN buys_computer = “no”
- IF age = “<=30” AND student = “yes” THEN buys_computer = “yes”
- IF age = “31..40” THEN buys_computer = “yes”
- IF age = “>40” AND credit_rating = “excellent” THEN buys_computer = “no”
- IF age = “>40” AND credit_rating = “fair” THEN buys_computer = “yes”

Bayes Classification

- What is Bayes Theorem?
  - States the probability of an event using prior knowledge of the condition that might affect the event.
  - It finds conditional probability.

Given a hypothesis H and evidence E , Bayes' theorem states that the relationship between the probability of the hypothesis before getting the evidence P(H) and the probability of the hypothesis after getting the evidence P(H|E) is

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

Image source : Google

Bayes Classification

Likelihood
How probable is the evidence
Given that our hypothesis is true?

Prior
How probable was our hypothesis
Before observing the evidence?

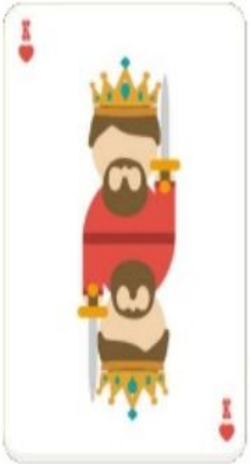
$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

Posterior
How probable is our Hypothesis
Given the observed evidence?
(Not directly computable)

Marginal
How probable is the new evidence
Under all possible hypothesis?

Image source : Google

Bayes Classification



$$P(King) = 4/52 = 1/13$$

$$P(King|Face) = \frac{P(Face|King).P(King)}{P(Face)}$$

$$P(Face|King) = 1$$

$$P(Face) = 12/52 = 3/13$$

$$= \frac{1.(1/13)}{3/13} = 1/3$$

Image source : Google

Rule-Based Classification

- Uses IF-THEN for classification purpose.
- IF condition THEN conclusion.
- Consider a rule R1
  - R1: IF age = youth AND student = yes THEN buy computer = yes
- IF part - rule precondition.
- THEN part - rule conclusion.
- Precondition (IF) part - one or more test attributes which are logically ANDed.
- Conclusion (THEN) part - class prediction.
- Other form of R1
- R1: (age = youth) ^ (student = yes))(buys computer = yes)

Image source : Google

Rule Extraction

- Extracting rules from decision tree.
  - One rule for each path from the root to the leaf node.
  - For rule precondition, logically AND splitting criterion .
  - Leaf node - class prediction.
- Rules
  - IF age = “<=30” AND student = “no” THEN buys computer = “no”
  - IF age = “<=30” AND student = “yes” THEN buys computer = “yes”
  - IF age = “31..40” THEN buys computer = “yes”
  - IF age = “>40” AND credit rating = “excellent” THEN buys computer = “no”
  - IF age = “>40” AND credit rating = “fair” THEN buys computer = “yes”

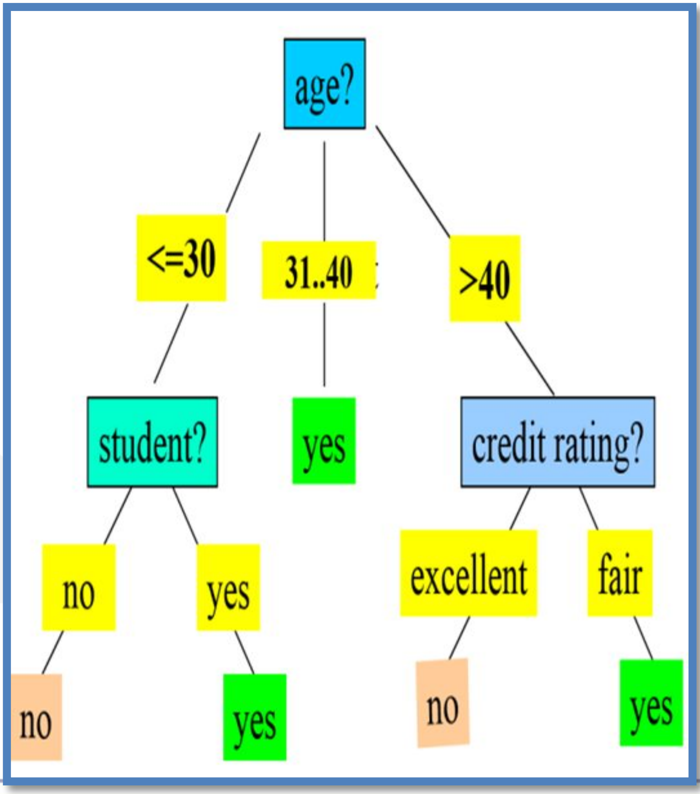


Image source : Google

Model Evaluation

- **For measuring accuracy of model, evaluation matrices are used.**
  - Tuples with class labeled
  - Validation test set
- **Methods**
  - Holdout
  - Random Sampling
  - Cross Validation
  - Bootstrap

Evaluation Matrix

- **Confusion Matrix.**

	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

- True Positive(TP) – Positive samples, predicted positive.

- False Positive(FP) – Negative samples, predicted positive.

- False Negative(FN) – Positive samples, predicted negative.

- True Negative(TN) – Negative samples, predicted negative.

Evaluation Measures

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Precision

$$\frac{TP}{TP + FP}$$

Recall

$$\frac{TP}{TP + FN}$$

F-measure

$$\frac{2 * Recall * Precision}{Recall + Precision}$$

- High recall, correctly classified class
- Lack positive examples but those classified positive are actually positive.
- F-measure uses harmonic mean.

Methods

- **Holdout.**
  - Random partitioning of the data
  - 2/3 data – training set
  - 1/3 – testing set
- **Random Sampling.**
  - Different version of holdout
  - Repeating holdout method k times
  - Final accuracy – avg accuracy of each iteration.

Methods

- **Cross Validation.**
  - Data splitting randomly in k –subsets of same size.
  - Use k-1 subsets as training data and remaining as testing data.
  - Repeat till every k subset is used as testing data.
- **Bootstrap.**
  - Sampling training tuples with replacement.
  - Among various bootstrap methods, 0.632 bootstrap is widely used.
  - Data set with d tuples sampled d times.
  - Tuples not covered in the training set forms the testing set.

Model Selection

- **Estimating Confidence Interval.**
  - Identify difference in the mean error rate of two different models using some statistical significance test.
  - Use 10 fold cross validation.
  - Assuming sample follows t distribution.
  - Hypothesis testing using t-test.
  - Hypothesis(Null Hypothesis) – two models are same or mean error difference between two is zero.
  - If the above hypothesis can be rejected than the conclusion is there is statistical significant difference between two models.
  - Choosing the model with lower rate.

Model Selection

- **ROC Curves.**
  - Receiver Operating Characteristics – shows performance of classifier model for all classification thresholds.
  - Accuracy of the model – area under ROC.
  - Positive class tuple ranks on top.
  - Demonstrate trade off between TPR and
  - FPR.
  - Perfect model – area = 1.0

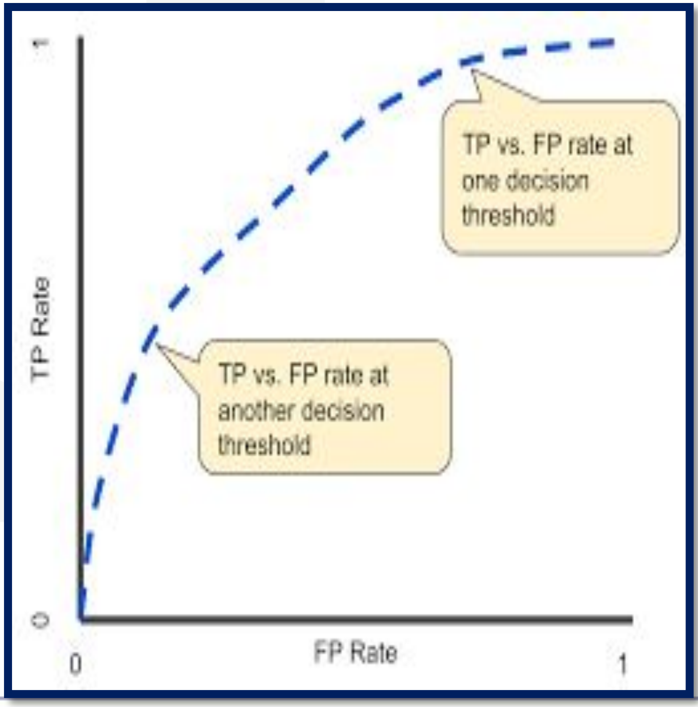


Image source : Google

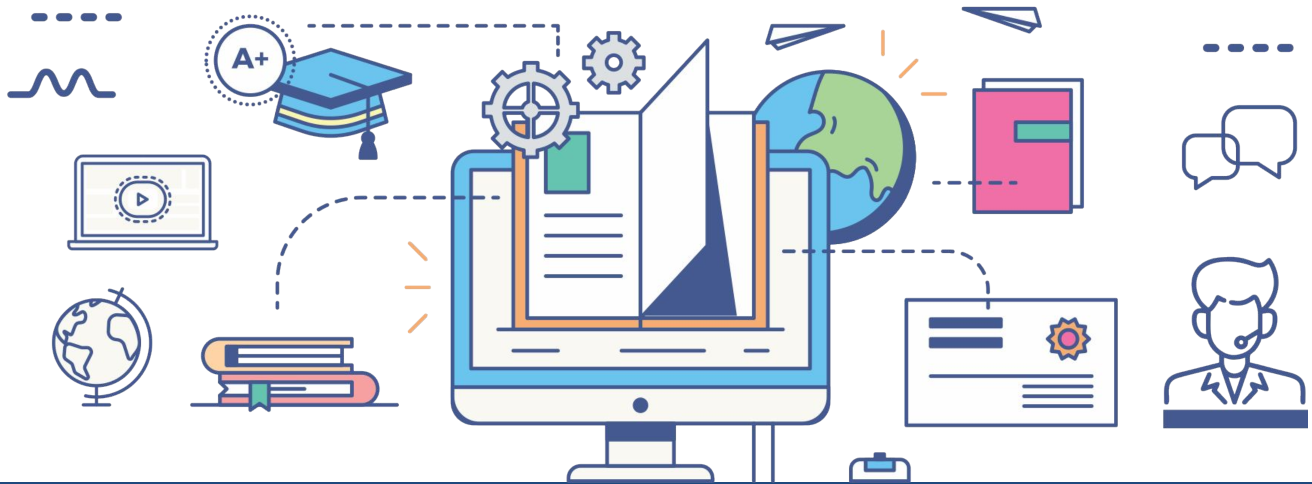
Evaluation of Rules

- Class labelled data set D. R is the rule defined.
  - ncovers - tuples covered by R
  - ncorrect - tuples correctly classified by R
- Assessment of R using coverage and accuracy measures.
  - coverage(R) = ncovers /|D|
  - accuracy(R) = ncorrect / ncovers

Advanced Classification Methods

Logistic Regression	• Probability of possible outcome. • Impact of independent variable on output.
K- Nearest Neighbours	• Classification based on majority voting • Efficient for large training data.
Random Forest	• Uses several decision trees on sub set of data. • Reduces overfitting.
Support Vector Machine	• Represents data as points in space separated by categories • Mapping of new samples in same space.

DIGITAL LEARNING CONTENT



Parul® University