Business Analytics

Today objective

Supervised Learning

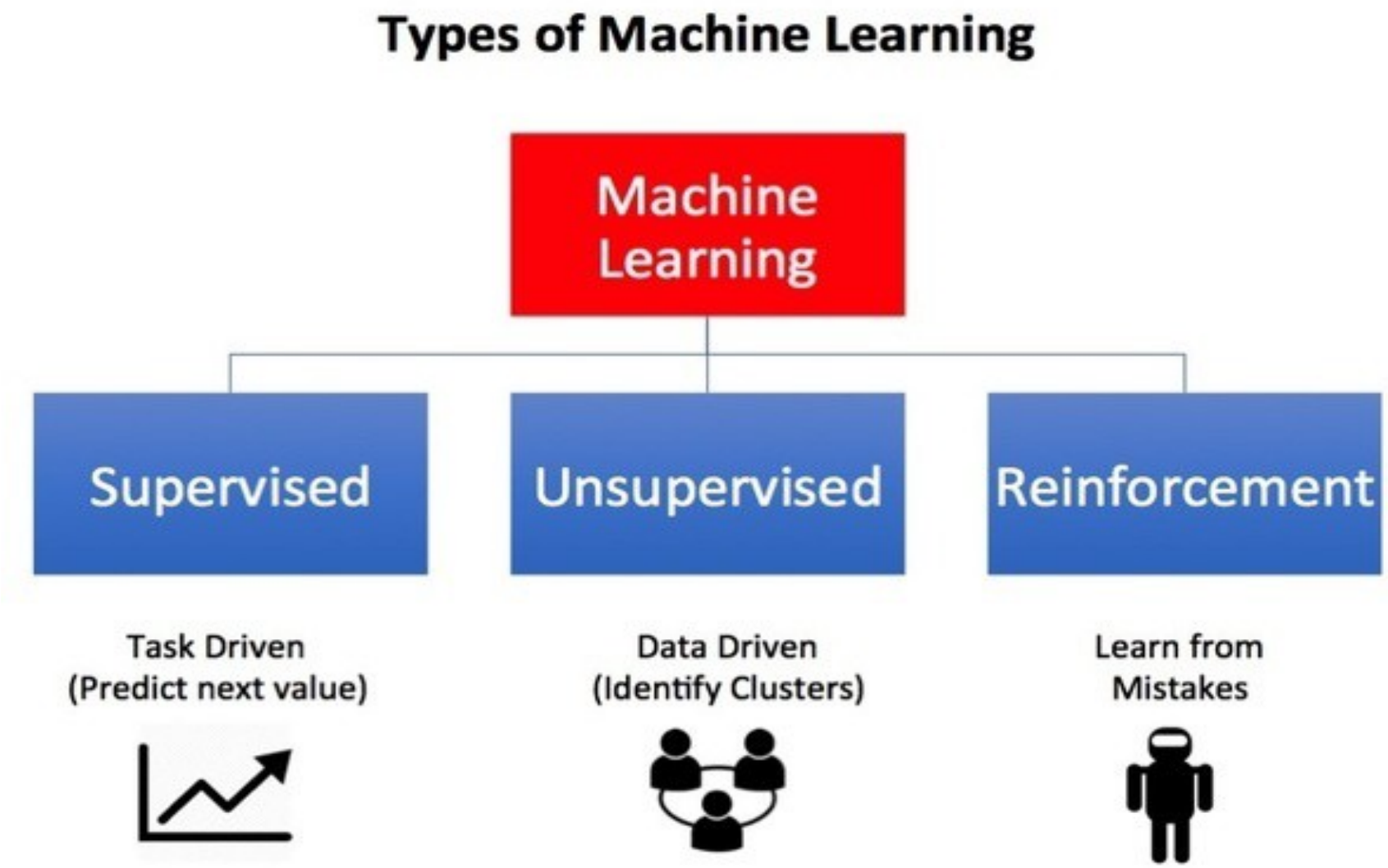
Machine Learning Approach

Predictive Analysis

Classification analysis

Decision Tree based Approach

Machine Learning



Classification

Classification: Use Cases

To find whether an email received is a spam or ham

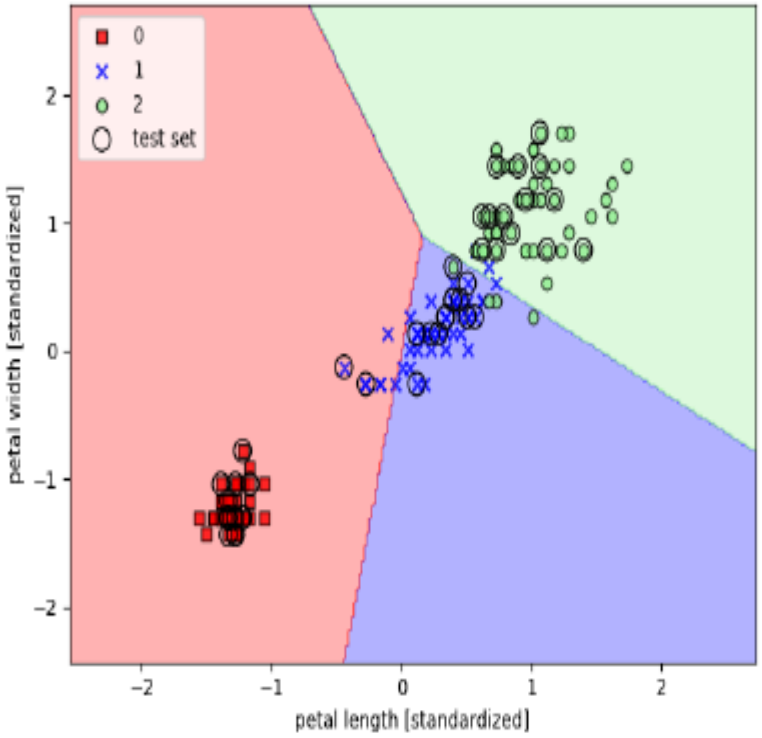
To identify customer segments

To find if a bank loan is granted

To identify if a kid will pass or fail in an examination

Classification

Classification: Example



- This chart shows classification of the Iris flower dataset into its three sub-species indicated by codes 0, 1, and 2.
- The test set dots represent assignment of new test data points to one class or the other based on the trained classifier model.

Classification of machine learning algorithms

Machine Learning Algorithms *(sample)*

	Unsupervised	Supervised

Continuous	• Clustering & Dimensionality Reduction
	  ○ SVD
	  ○ PCA
	  ○ K-means
		• Regression
		  ○ Linear
		  ○ Polynomial
		• Decision Trees
		• Random Forests

Categorical	• Association Analysis
	  ○ Apriori
	  ○ FP-Growth
	• Hidden Markov Model
		• Classification
		  ○ KNN
		  ○ Trees
		  ○ Logistic Regression
		  ○ Naive-Bayes
		  ○ SVM

Classification

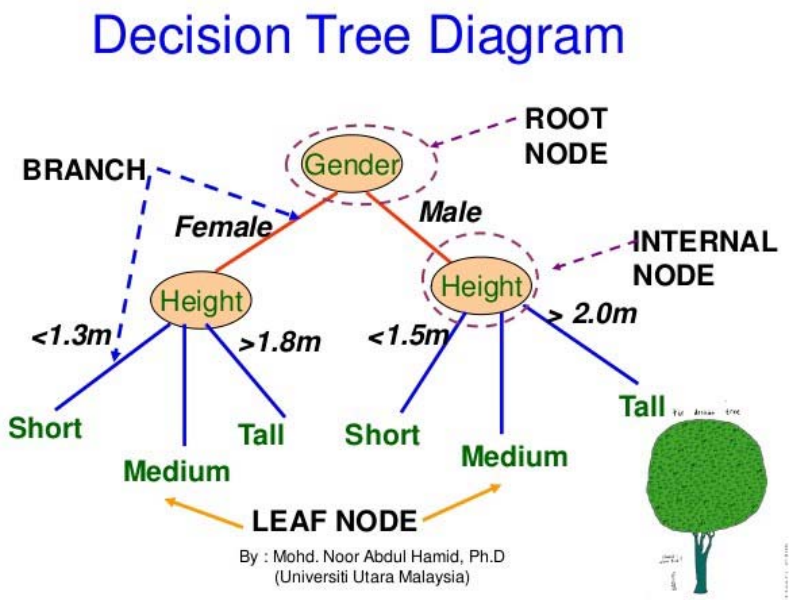
Decision Trees are used to predict a Label (usually Binary) dependent variables such as:

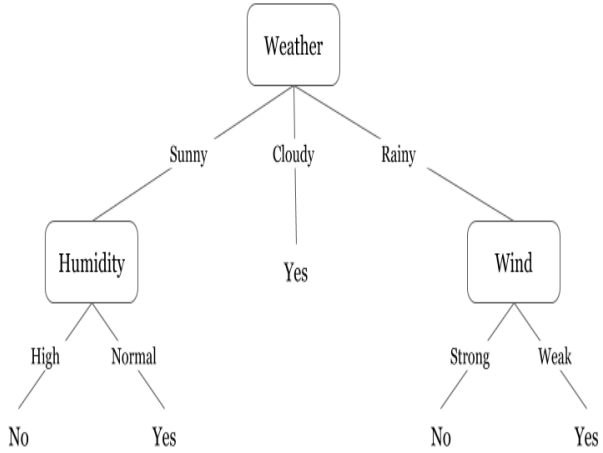
- Will a person suffer a heart attack in the next year?
- Will a voter vote BJP in the next country election?
- Will student X clear this time IAS exam?

For such type of problem ,decision tree gives basic solution based upon past data.

Decision Tree

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand.





Classification
(a decision tree structure)

- 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
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Confusion Matrix

What is a Confusion Matrix?

The million-dollar question – what, after all, is a confusion matrix?

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:

		Classifier Prediction: Positive	Classifier Prediction: Negative
Actual Value	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

A couple other terms are also worth mentioning:

•**Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance.

•**F Score:** This is a weighted average of the true positive rate (recall) and precision.

•**ROC Curve:** This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.

•**Accuracy:** Overall, how often is the classifier correct?

•(TP+TN)/total = (100+50)/165 = 0.91

•**Misclassification Rate:** Overall, how often is it wrong?

•(FP+FN)/total = (10+5)/165 = 0.09

•equivalent to 1 minus Accuracy

•also known as "Error Rate"

•**True Negative Rate:** When it's actually no, how often does it predict no?

•TN/actual no = 50/60 = 0.83

•equivalent to 1 minus False Positive Rate

•also known as "Specificity"

•**True Positive Rate:** When it's actually yes, how often does it predict yes?TP/actual

•yes = 100/105 = 0.95

•also known as "Sensitivity" or "Recall"

•**False Positive Rate:** When it's actually no, how often does it predict yes?FP/actual no = 10/60 = 0.17

•**Precision:** When it predicts yes, how often is it correct?

•TP/predicted yes = 100/110 = 0.91

•**Prevalence:** How often does the yes condition actually occur in our sample?

•actual yes/total = 105/165 = 0.64

Classification
(a decision tree structure)

- 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
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

In decision tree learning, **ID3 (Iterative Dichotomiser 3)** is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset. ID3 typically used in the machine learning and natural language processing domains.

Procedure for Decision Tree Induction

- Basic Procedure
  - Tree is constructed in a top-down recursive 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

Attribute Selection Measure: Information Gain(ID3)

- 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 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 by branching on attribute A

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

Attribute Selection: Information Gain

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

9 y and 5 no

$$Info(D) = I(y,n) = -\frac{y}{total}\log_2(\frac{y}{total}) - \frac{n}{total}\log_2(\frac{n}{total}) = value$$

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

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

Attribute Selection: Information Gain

$$Info(D) = I(y,n) = -\frac{y}{total}\log_2(\frac{y}{total}) - \frac{n}{total}\log_2(\frac{n}{total}) = value$$

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

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

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

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

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

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

Attribute Selection: Information Gain

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

$$Gain(income) = 0.029$$

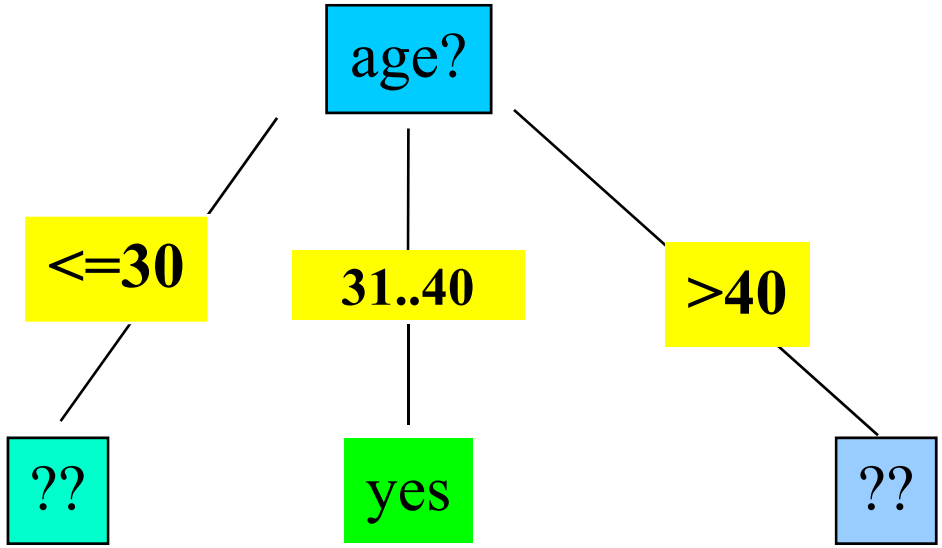
$$Gain(student) = 0.151$$

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

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

Max gain

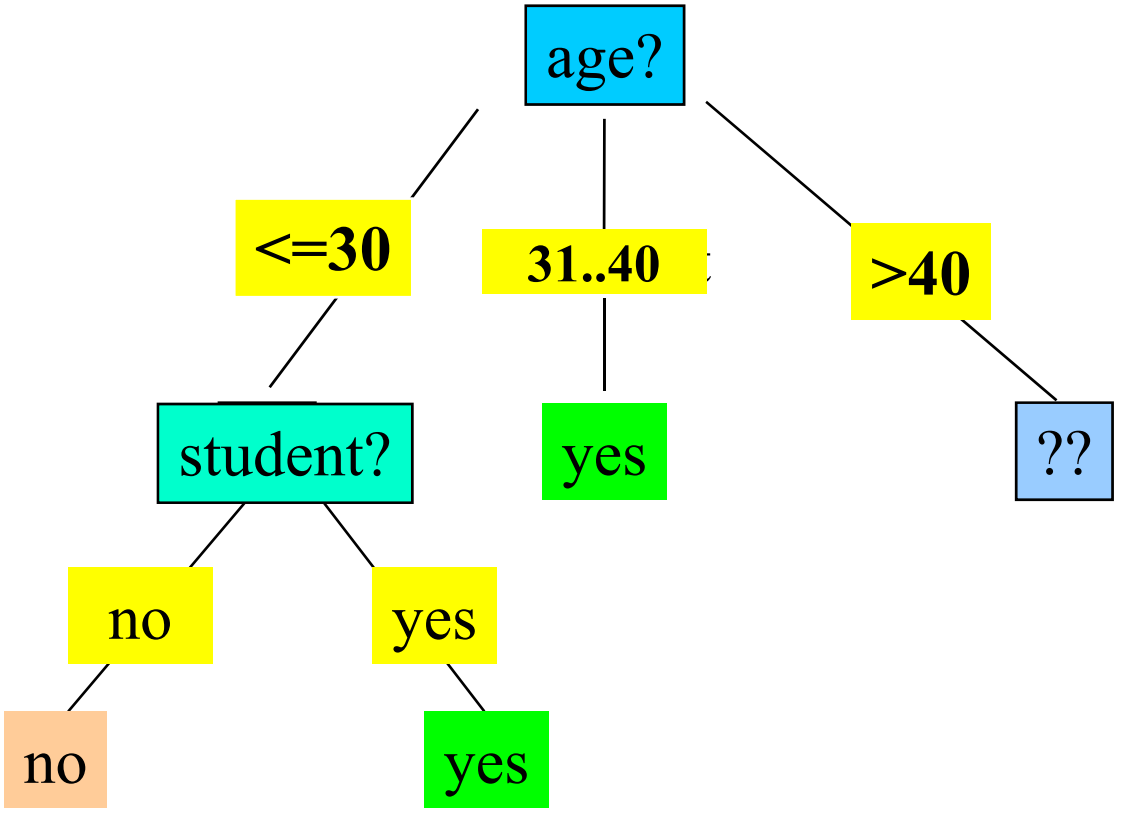
Age



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

For <=30

Gain(Student) =1 Gain(C_R) =.05 Gain(Income) =.6



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

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

For >40

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

Credit_rating	pi	n_i	$I(p_i, n_i)$
Fair	3	0	
Excellent	0	2	

$$Info(D) = I(5,5) = -\frac{5}{10}\log_2(\frac{5}{10}) - \frac{5}{10}\log_2(\frac{5}{10}) = 1$$

$$Info_{student}(D) = \frac{3}{5} I(3,0) + \frac{2}{5} I(0,2) = .0$$

Gain(C_R) =1-0=1

For >40

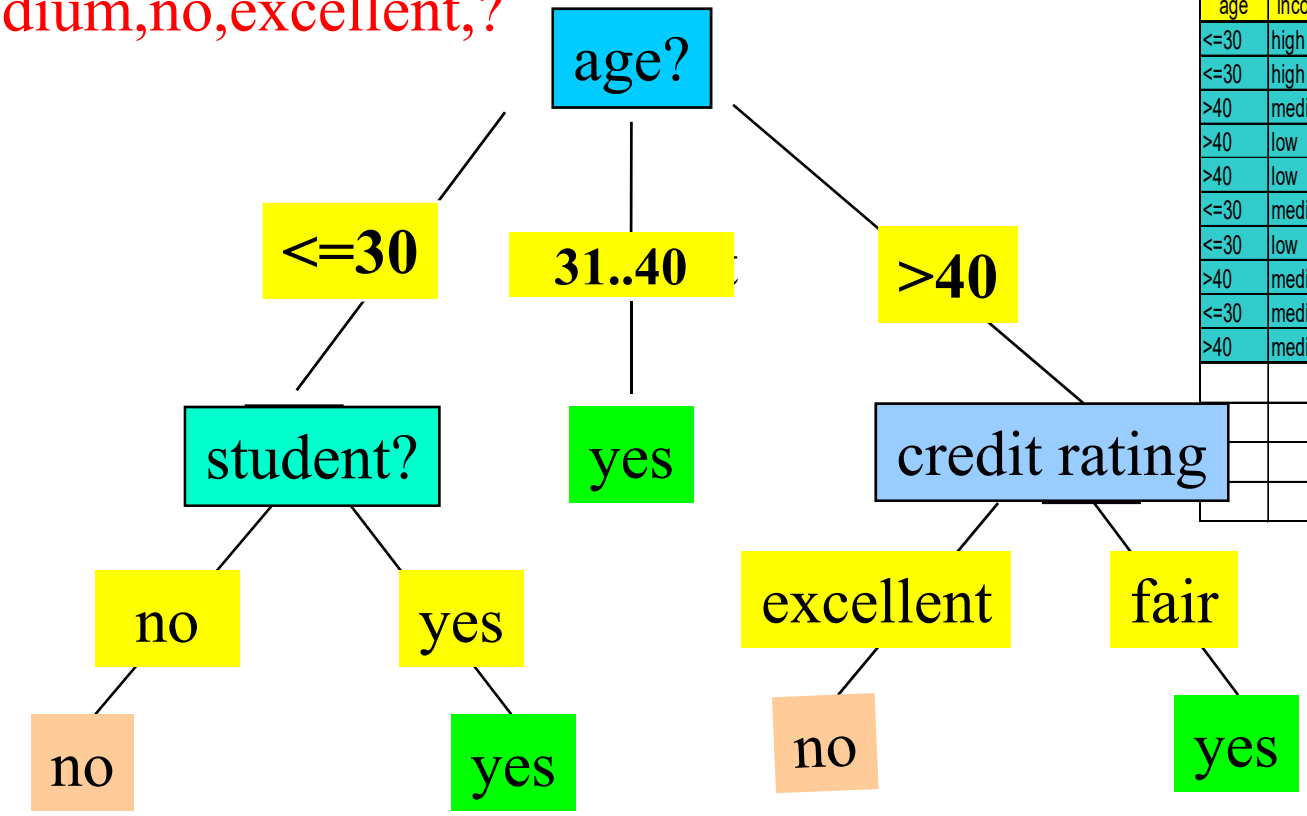
Gain(Student) =1-.95=.05

Gain(Income) =1-.95=.05

Gain(C_R) =1-0=1

New Data??

>40,medium,no,excellent,?



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

Simulation using R

The *iris* dataset

Iris Versicolor

Iris Setosa

Iris Virginica

- This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 iris. The are *Iris setosa*, *versicolor*, and *virginica*.
- Based on the variables sepal length and width and petal length and width, we have to classify the flowers as *Iris setosa*, *versicolor*, and *virginica*

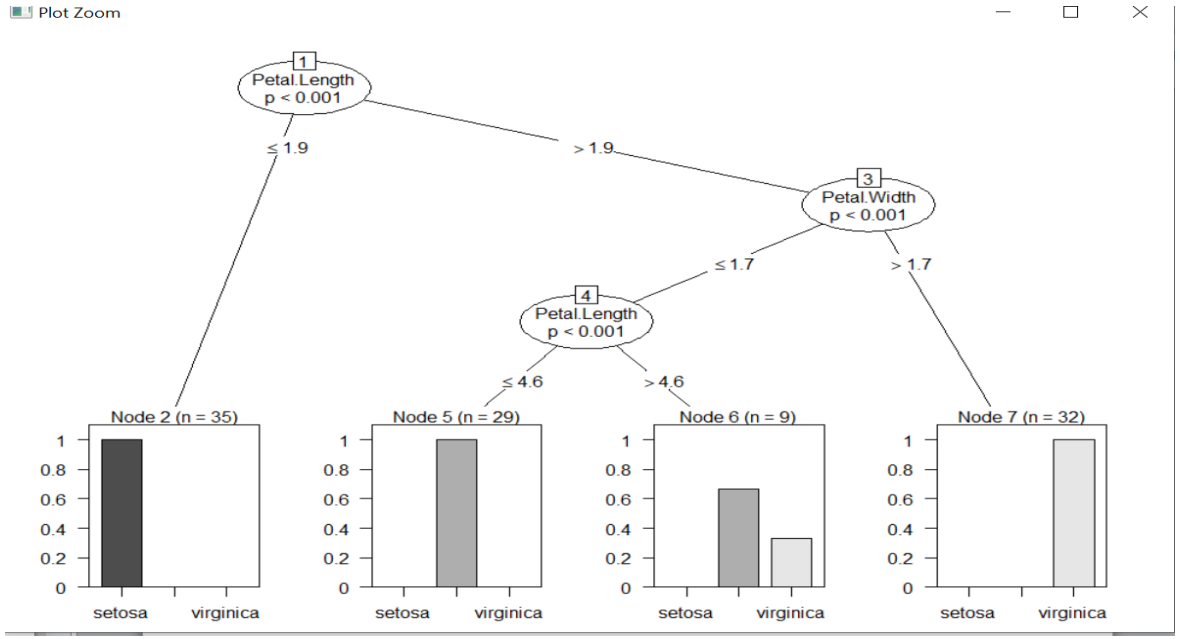
Splitting into training and test dataset

```
library(caTools)

split <- sample.split(iris$Species, SplitRatio = 0.7)
train = subset(iris, split == TRUE)
test = subset(iris, split == FALSE)
```

R code to run decision tree model

```
library("party")
m <- ctree(Species ~ ., data = train)
```



Thank you !!!