Module VI

CHAPTER 6

Introduction to ML

University Prescribed Syllabus w.e.f Academic Year 2021-2022

Introduction to Machine Learning,Types of Machine Learning : Supervised (Logistic Regression, Decision Tree,Support Vector Machine) and Unsupervised (K Means Clustering,Hierarchical Clustering,Association Rules) Issues in Machine Learning, Applications of Machine Learning, Steps in developing a Machine Learning Application.

Self-Learning Topics : Real world case studies on machine learning

Teaching Hours – 04 **Approximate Weightage of Marks in Exam. – 10 Marks**

6.1 INTRODUCTION TO MACHINE LEARNING

UQ. What is Machine learning ?
(MU - May 17, 2 Marks May 19, 5 Marks)

UQ. Define Machine learning and explain with example importance of Machine Learning.
(MU - Dec. 19, 5 Marks)

- A **machine** that is intellectually capable as much as humans, have always attracted writers and early computer scientist who were excited about artificial intelligence and machine learning.
- The first machine learning system was developed in the 1950s. In 1952, Samuel has developed a program to play checkers. The program was able to observe positions at game and learn the model that gives better moves for machine player.
- In 1957, Frank Rosenblatt designed the Perceptron, which is a simple classifier but when it is combined in large numbers, in a network, it became a powerful tool.
- Minsky in 1960, came up with limitation of perceptron. He showed that the X-OR problem could not be represented by perceptron and such inseparable data distribution cannot be handled and following this Minsky's work neural network research went to dormant until 1980s.
- Machine learning became very famous in 1990s,due to the introduction of statistics. Computer science and statistics combination lead to probabilistic approaches in Artificial intelligence.
- This area is further shifted to data driven techniques. As Huge amount of data is available, scientists started to design intelligent systems that are able to analyze and learn from data.
- Machine learning is a category of Artificial Intelligence. In machine learning computers has the ability to learn themselves, explicit programming is not required.
- Machine focuses on the study and development of algorithms that can learn from data and also make predictions on data.
- Machine learning is defined by Tom Mitchell as "A program learns from experience 'E' with respect to some class of tasks 'T' and performance measure 'P', if its performance on tasks in 'T' as measured by 'P' improves with 'E'." Here 'E' represents the past experienced data and 'T' represents the tasks such as prediction, classification, etc. Example of 'P', we might want to increase accuracy in prediction.
- Machine learning mainly focuses on the design and development of computer programs that can teach themselves to grow and change when exposed to new data.
- Using machine learning we can collect information from a dataset by asking the computer to make some sense from data. **Machine learning** is turning data into information.

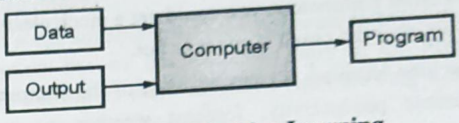


Fig. 6.1.1 : Machine Learning

- The Fig. 6.1.2 is the schematic representation of the Machine Learning system. ML system takes the training data and background knowledge as the input.
- Background knowledge and data helps the Learner program to provide a solution for a particular task or problem.
- Performance corresponding to the solution can be also measured. ML system comprises of mainly two components, Learner and a Reasoner.
- Learner use the training data and background knowledge to build the model and this can be used by reasoner to provide the solution for a task.
- Machine learning can be applied to many applications such as politics to geosciences. It is a tool that can be applied to many problems.
- Any application which needs to extract some information from data and also takes some action on data, can benefit from machine learning methods.
- Some of the applications are spam filtering in email, face recognition, product recommendations from Amazon.com and handwriting digit recognition.

Module 6

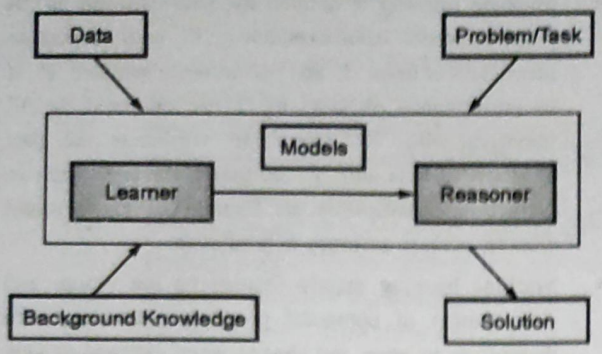


Fig. 6.1.2 : Schematic diagram of Machine Learning

- In detecting spam email, if you check for the occurrence of single word it will not be very helpful.
- But checking the occurrences of certain words used together and combined this with the length of the email and other parameters, you could get a much clearer idea of whether the email is spam or not.
- Machine learning is used by most of the companies to increase productivity, forecast weather, to improve business decisions, detect disease and do many more things.
- Machine learning uses statistics. There are many problems where the solution is not deterministic. There are certain problems for which we don't have that much information and also don't have that much computing power to properly model the problem.
- For these problems we need statistics, example of such type of problem is prediction of motivation and behavior of humans.
- The behavior and motivation of humans is a problem that is currently very difficult to model.

Machine learning = Take data + understand it + process it + extract value from it + visualize it + communicate it

6.2 KEY TERMINOLOGY

UQ. What are the key tasks of Machine Learning ?
(MU - May 16, 5 Marks)

☞ **Expert System**

- Expert system is a system which is developed using some training set, testing set, and knowledge representation, features, algorithm and classification terminology.

(i) **Training Set :** A training set comprises of training examples which will be used to train machine learning algorithms.

(ii) **Testing Set :** To test machine learning algorithms what's usually done is to have a training set of data and a separate dataset, called a test set.

(iii) **Knowledge Representation :** Knowledge representation may be stored in the form of a set of rules. It may be an example from the training set or a probability distribution.

(iv) **Features :** Important properties or attributes.

(v) **Classification :** We classify the data based on features.

- **Process :** Suppose we want to use a machine learning algorithm for classification. The next step is to train the algorithm, or allows it to learn. To train the algorithm we give as a input a quality data called as **training set.**
- Each training example has some features and one target variable. The target variable is what we will be trying to predict with our machine learning algorithms.
- In a training dataset the target variable is known. The machine learns by finding some relationship between the target variable and the features.
- In the classification tasks the target variables are known as classes. It is assumed that there will be a limited number of classes.
- The class or target variable that the training example belongs to is then compared to the predicted value, and we can get a idea about the accuracy of the algorithm.
- **Example :** First we will see some terminologies that are frequently used in machine learning methods. Let's take an example that we want to design a classification system that will classify the instances in to either Acceptable or Unacceptable. This kind of system is a fascinating topic often related with machine learning called **expert systems.**
- Four features of the various cars are stored in Table 6.2.1. The features or the attributes selected are Buying_Price, Maintenance_Price, Lug_Boot and Safety. Examples belong to Table 6.2.1 represents a record comprises of features.

- In Table 6.2.1 all the features are categorical in nature and takes limited disjoint values. The first two features represent the buying price and maintenance price of a car such as high, medium and low.
- Third feature shows the luggage capacity of a car as small, medium or big. Fourth feature represents whether the car has safety measures or not, which takes the value as low, medium or high.
- Classification is one of the important task in machine learning. In this application we want to evaluate the car out of a group of other cars. Suppose we have all information about car's Buying_Price, Maintenance_Price, Lug_Boot and Safety.
- Classification method is used to evaluate a given car as Acceptable or Unacceptable. Many machine learning algorithms are there that can be used for classification. The target or the response variable in this example is the evaluation of a car.
- Suppose we have selected a machine learning algorithm to use for classification. The main task in the classification is to train the algorithm, or allow it to learn. We give the experienced data as the input to train the algorithm which is called as training data.
- Let's assume training dataset contains 14 training records in Table 6.2.1. Suppose each training record has four features and one target or the response variable, as shown in Fig. 6.2.1. The machine learning algorithm is used to predict the target variable.
- In classification task the target variable takes a discrete value, and in the task of regression its value could be continuous.
- In a training dataset we have the value of target variable. The relationship that exists between the features and the target variable is used by machine for learning.
- The target variable is the evaluation of the car. Classes are the target variables in the classification task. In classification systems it is assumed that classes are to be of limited number.
- Attributes or features are the individual values that, when combined with other features, make up a training example. This is usually columns in a training or test set. A training dataset and a testing dataset, is used to test machine learning algorithms. First the training dataset is given as input to the program.
- Program uses this data to learn. Next, the test set is given to the program. The program decides which instance of test data belongs to which class.
- The predicted output is compared with the actual output of the program, and we can get an idea about the accuracy of the algorithm. There are best ways to use all the information in the training dataset and test dataset. Assume in car evaluation classification system, we have tested the program and it meets the desired level of accuracy.
- Knowledge representation is used to check what the machine has learned. There are many ways in which knowledge can be represented.
- We can use set of rules or a probability distribution to represent the knowledge. Many algorithms represent the knowledge which is more interpretable to humans than others.
- In some situations we may not want to build an expert system but we are interested only in the knowledge representation that's acquired from training a machine learning algorithm.

Table 6.2.1 : Car evaluation classification based on four features

Buying_Price	Maintenance_ Price	Lug_Boot	Safety	Evaluation?
High	High	Small	High	Unacceptable
High	High	Small	Low	Unacceptable
Medium	High	Small	High	Acceptable
Low	Medium	Small	High	Acceptable
Low	Low	Big	High	Acceptable
Low	Low	Big	Low	Unacceptable
Medium	Low	Big	Low	Acceptable
High	Medium	Small	High	Unacceptable
High	Low	Big	High	Acceptable
Low	Medium	Big	High	Acceptable
High	Medium	Big	Low	Acceptable
Medium	Medium	Small	Low	Acceptable
Medium	High	Big	High	Acceptable
Low	Medium	Small	Low	Unacceptable

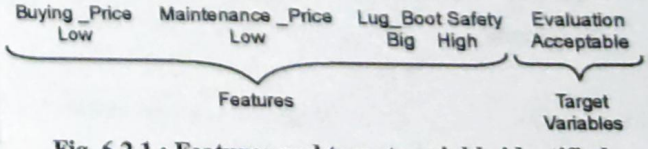


Fig. 6.2.1 : Features and target variable identified

6.3 TYPES OF MACHINE LEARNING

UQ. Explain how supervised learning is different from unsupervised learning. **(MU - May 17, 3 Marks)**

UQ. What are main types of Machine Learning ? **(MU - May 16, 5 Marks)**

Some of the main types of machine learning are:

(1) Supervised Learning

- In this type of learning we use data which is comprises of input and corresponding output. For every instance of data we can have input 'X' and corresponding output 'Y'.
- From this ML system will build model so that given an observation 'X', for new observation 'X' it will try to find out what is corresponding 'Y'.
- In supervised learning training data is labelled with the correct answers, e.g. "spam" or "ham." Two most important types of supervised learning are **classification** (where the outputs are discrete labels, as in spam filtering) and **regression** (where the outputs are real-valued).

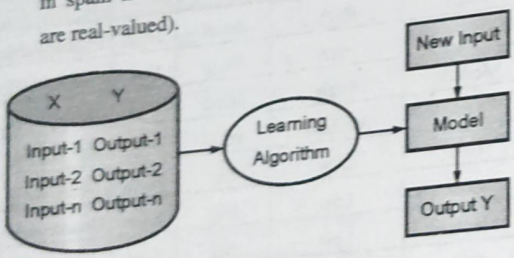


Fig. 6.3.1 : Supervised Learning

(2) Unsupervised learning

- In unsupervised learning you are only given input 'X', there is no label to the data and given the data or different data points, you may want to form clusters or want to find some pattern. Two important unsupervised learning tasks are **dimension reduction** and **clustering**.

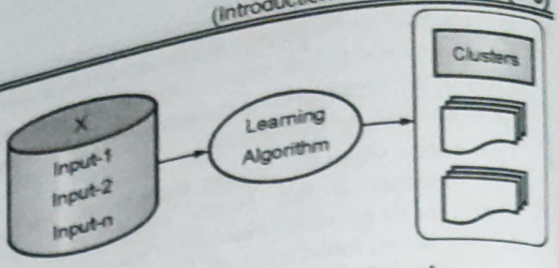


Fig. 6.3.2 : Unsupervised Learning

(3) Reinforcement learning

- In reinforcement learning you have an agent who is acting in an environment and you want to find out what action the agent must take based on the reward or penalty that the agent gets it. In this an agent (e.g., a robot or controller) seeks to learn the optimal actions to take based the outcomes of past actions.

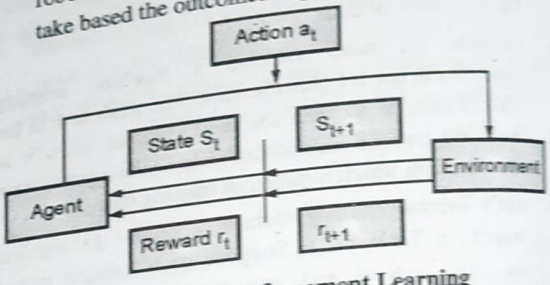


Fig. 6.3.3 : Reinforcement Learning

(4) Semi-supervised learning

- It is a combination of supervised and unsupervised learning. In this there is some amount of labeled training data and also you have large amount of unlabeled data and you try to come up with some learning algorithm that convert even when training data is not labeled.
- In classification task, the aim is to predict class of an instance of data. Another method in machine learning is regression.
- Regression is the prediction of a numeric value. Regression's example is to draw a best fit line which passes through some data points in order to generalize the data points.
- Classification and regression are examples of supervised learning. These types of problems are called as supervised because we are asking the algorithm what to predict.

- The exact opposite of supervised is a task called as unsupervised learning. In unsupervised learning, target value or label is not given for the data.
- A problem in which similar items are grouped together is called as clustering. In unsupervised learning, we may also want to find statistical values that describe the data.
- This is called as **density estimation**. Another task of unsupervised learning may be reducing the huge amount of data from many attributes to a small number so that we can properly visualize it in two or three dimensions.

Table 6.3.1 : Supervised learning tasks

k-Nearest Neighbours	Linear
Naïve Bayes	Locally weighted linear
Support Vector Machines	Ridge
Decision Trees	Lasso

Table 6.3.2 : Unsupervised learning tasks

DBSCAN	Parzen Window
k-Means	Expectation Maximization

6.4 SUPERVISED LEARNING : LOGISTIC REGRESSION

UQ. Write short note on : Logistic Regression.
(MU - May 17, 10 Marks)

- Suppose we have different training data which belongs to two different classes, and we have to design a system that will identify which data is from which class.
- The output of this function will be a real value and it is not suitable for classification method. We can use another function on this linear function so that we can use the result for classification.
- In logistic regression logistic function or the sigmoid can be used. Let's first see what is the meaning of Logistic or Sigmoid function.
- In Logistic regression classifier we will take features as the input. These features are multiplied with the logistic coefficients and we add the product. This is called as the net input that will be given to the sigmoid function.
- The output will be between 0 and 1. Anything above 0.5 is classified as 1and anything below 0.5 is classified as 0.

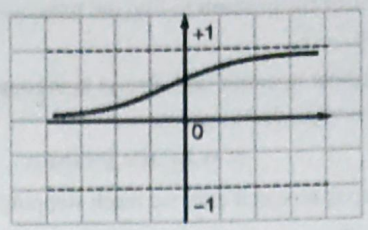


Fig. 6.4.1 : Logistic or Sigmoid Function

- The net input to the sigmoid function is,

$$Z = b_0 + b_1 x_1 + b_2 x_2 + \ldots\ldots + b_n x_n$$

- In the net input equation x represents the input data or the features. When we use optimized coefficient b, Classifier will be successful. Optimized b can be calculated using the optimization concept.

6.4.1 Gradient Ascent Method

- In gradient ascent method we move in the direction of the gradient to find the maximum point on a function.

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

- In the above equation $\partial f(x, y) / \partial x$ represents the amount by which updation is applied in x direction and $\partial f(x, y) / \partial y$ represents the amount by which updation applied in y direction
- The gradient operator will always point towards the direction of gradient increase.

$$b = b + \alpha \times \nabla f(b)$$

- This step is repeated until we reach stopping criterion.

Pseudo code

```
Start using the logistic coefficient, all set to 0
Repeat no. of times
  Find the gradient of complete dataset
  Update, logistic coefficient = logistic coefficient + alpha ×
                                 gradient
  Return the logistic coefficient
```

Module 6

6.4.2 Gradient Descent Method

- In gradient descent method we move in the opposite direction of the gradient to find the minimum point on a function
- The gradient operator will always point opposite to the direction of gradient increase.

$$b = b - \alpha \times \nabla f(b)$$

- This step is repeated until we reach stopping criterion.
- Using above mentioned methods optimized b is calculated, net input is calculated and then the prediction is calculated by giving the net input to the function,

$$\text{Prediction} = \frac{1}{1+e^{-x}}$$

- Finally the data points are classified as,

$$\text{Class} = 1 \quad \text{if Prediction} \geq 0.5$$
$$= 0 \quad \text{else}$$

- The classified data using the decision boundary is as shown in Fig. 6.4.2.
- Logistic regression is used to model a relationship between input variables and a target variable. Let's take an example of a house management system , we can use logistic regression to model the relationship between the parameters such as the total monthly income of the family, various liabilities, monthly expenditure of a family to predict if monthly savings (investment) can be done or not.

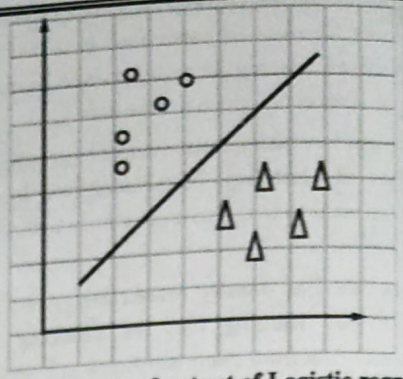


Fig. 6.4.2 : Sample of output of Logistic regression

6.4.3 Types of Logistic Regression

There are three types of logistic regression as below,

1. **Binary Logistic Regression**

- Binary logistic regression is used if the target variable is binary.
- The application of this method can be above mentioned house management example.

2. **Nominal Logistic Regression**

- Nominal Logistic Regression is used if there are three or more classes without any specific orders.
- The examples of this type of regression could be different departments of the engineering college (Computer, IT, Mechanical, Civil, Electronics etc.).

3. **Ordinal Logistic Regression**

- Ordinal Logistic Regression is used if there are three or more classes with specific orders.
- The examples of this type of regression could be how customers rate the taste of food using a scale of 1-3 (Bad, Good, and Excellent).

6.4.4 Examples on Logistic Regression

Ex. 6.4.1 : A Bank has to decide whether to sanction loan or not based on two attributes as person's income and his savings. The data is given in the following table where 1 represents loan is sanctioned and 0 represents loan is not sanctioned. Predict whether a person 3 will get a loan or not having annual income as 12.5 lakhs and savings as 10 lakhs.

Person	Annual income in lakhs (x_1)	Savings in lakhs (x_2)	Loan sanctioned? (y)
1	14.5	12.5	1
2	8.5	4.5	0

☑ **Soln. :**

Initially assume logistic regression coefficients $b_0 = b_1 = b_2 = 0$

For 1st row, $x_1 = 14.5$, $x_2 = 12.5$ and $y = 1$

Now we will calculate prediction for the first row,

$$\text{Prediction} = 1 / (1 + e^{\wedge} (- (b_0 + b_1 \times x_1 + b_2 \times x_2)))$$

$$\text{Prediction} = 1 / (1 + e^{\wedge} (- (0 + 0 \times 14.5 + 0 \times 12.5)))$$

$$\text{Prediction} = 0.5$$

Now we will calculate the new coefficient values using a simple update equation. Ideal values for alpha are from 0.1 to 0.3. Let's take alpha as 0.3. For b_0 by default input is 1.

$$b_{new} = b_{old} + \text{alpha} \times (y - \text{prediction}) \times \text{prediction} \times (1 - \text{prediction}) \times \text{input}$$

$$b_{0new} = 0 + 0.3 \times (1 - 0.5) \times 0.5 \times (1 - 0.5) \times 1.0 = 0.0375$$

$$b_{1new} = 0 + 0.3 \times (1 - 0.5) \times 0.5 \times (1 - 0.5) \times 14.5 = 0.54375$$

$$b_{2new} = 0 + 0.3 \times (1 - 0.5) \times 0.5 \times (1 - 0.5) \times 12.5 = 0.46875$$

Now we will calculate prediction for the second row,

$$\text{Prediction} = 1 / (1 + e^{\wedge} (- (b_0 + b_1 \times x_1 + b_2 \times x_2)))$$

$$\text{Prediction} = 1 / (1 + e^{\wedge} (- (0.0375 + 0.54735 \times 8.5 + 0.46875 \times 4.5)))$$

$$\text{Prediction} = 0.99$$

Now we will calculate the new coefficient values

$$b_{new} = b_{old} + \text{alpha} \times (y - \text{prediction}) \times \text{prediction} \times (1 - \text{prediction}) \times \text{input}$$

$$b_{0new} = 0.0375 + 0.3 \times (0 - 0.99) \times 0.99 \times (1 - 0.99) \times 1.0 = 0.034$$

$$b_{1new} = 0.54735 + 0.3 \times (0 - 0.99) \times 0.99 \times (1 - 0.99) \times 8.5 = 0.523$$

$$b_{2new} = 0.46875 + 0.3 \times (0 - 0.99) \times 0.99 \times (1 - 0.99) \times 4.5 = 0.456$$

Now we will use these values for prediction

$$\text{Prediction} = 1 / (1 + e^{\wedge} (- (b_0 + b_1 \times x_1 + b_2 \times x_2)))$$

$$\text{Prediction} = 1 / (1 + e^{\wedge} (- (0.034 + 0.523 \times 12.5 + 0.456 \times 10)))$$

$$\text{Prediction} = 0.99$$

$$\text{Since prediction} \geq 0.5$$

Prediction for person 3 is, his loan will be sanctioned.

Note: Generally huge amount of data is used for training and number of iterations are also applied to get accuracy. Here for example purpose only two records for training are taken and a single iteration is shown.

6.5 SUPERVISED LEARNING : DECISION TREE

6.5.1 Introduction to Decision Tree

UQ. Write short note on Issues in Decision Tree. **(MU - May 15, 10 Marks)**

- Decision trees are very strong and most suitable tools for classification and prediction. The attractiveness of decision trees is due to the fact that, in contrast to neural network, decision trees represent rules.
- Rules are represented using linguistic variables so that user interpretability may be achieved. By comparing the records with the rules one can easily find a particular category to which the record belongs to.
- In some applications, the accuracy of a classification or prediction is the only thing that matters in such situations we do not necessarily care how or why the model works.
- In other situations, the ability to explain the reason for a decision is crucial, in marketing one has described the customer segments to marketing professionals, so that they can use this knowledge to start a victorious marketing campaign.

Module 6

- This domain expert must acknowledge and approve this discovered knowledge and for this we need good descriptions. There are a variety of algorithms for building decision trees that share the desirable quality of interpretability (ID3).

1. Where Decision Tree is applicable ?

- Decision tree method is mainly used for the tasks that possess the following properties.
- The tasks or the problems in which the records are represented by attribute-value pairs.

  Records are represented by a fixed set of attributes and their value Example: For 'temperature' attribute the value is 'hot'. When there are small numbers of disjoint possible values for each attribute, then decision tree learning becomes very simple.

  Example : Temperature attribute takes three values as hot, mild and cold.

  Basic decision tree algorithm may be extended to allow real valued attributes as well.

  Example : We can define floating point temperature.
- An application where the target function takes discrete output values.

  In Decision tree methods an easiest situation exists, if there are only two possible classes.

  Example: Yes or No

  When there are more than two possible output classes then decision tree methods can also be easily extended.

  A more significant extension allows learning target functions with real valued outputs, although the application of decision trees in this area is not frequent.
- The tasks or the problems where the basic requirement is the disjunctive descriptors. Decision trees naturally represent disjunctive expressions.
- In certain cases where the training data may contain errors. Decision tree learning methods are tolerant to errors that can be a classification error of training records or attribute-value representation error.
- The training data may be incomplete as there are missing attribute values. Although some training records have unknown values, decision tree methods can be used.

2. Decision Tree Representation

- Decision tree is a classifier which is represented in the form of a tree structure where each node is either a leaf node or a decision node.
  - Leaf node represents the value of the target or response attribute (class) of examples.
  - Decision node represents some test to be carried out on a single attribute-value, with one branch and sub tree for each possible outcome of the test.
- Decision tree generates regression or classification models in the form of a tree structure. Decision tree divides a dataset into smaller subsets with increase in depth of tree.
- The final decision tree is a tree with decision nodes and leaf nodes. A decision node (e.g., Buying_Price) has two or more branches (e.g., High, Medium and Low). Leaf node (e.g., Evaluation) shows a classification or decision.
- The topmost decision node in a tree which represents the best predictor is called root node. Decision trees can be used to represent categorical as well as numerical data.
  - **Root Node :** It represents entire set of records or dataset and this is again divided into two or more similar sets.
  - **Splitting :** Splitting procedure is used to divide a node into two or more sub-nodes depending on the criteria.
  - **Decision Node :** A decision node is a sub-node which is divided into more sub-nodes.
  - **Leaf/ Terminal Node :** Leaf node is a node which is not further divided or a node with no children.
  - **Parent and Child Node :** Parent node is a node, which is split into sub-nodes and sub-nodes are called as child of parent node.
  - **Branch / Sub-Tree :** A branch or sub-tree is a sub part of decision tree.
  - **Pruning :** Pruning method is used to reduce the size of decision trees by removing nodes.

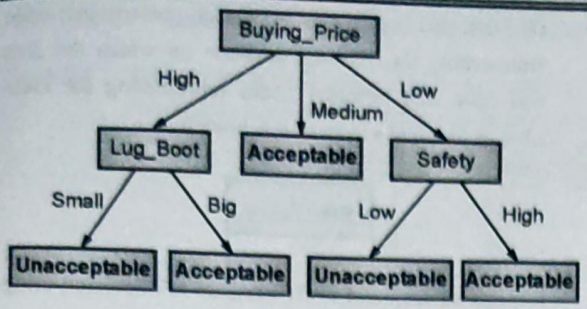


Fig. 6.5.1

3. Attribute Selection Measure

1. Gini Index

- All attributes are assumed to be continuous valued.
- It is assumed that there exist several possible split values for each attribute.
- Gini index method can be modified for categorical attributes.
- Gini is used in Classification and Regression Tree (CART).

If a data set T contains example from n classes, gini index, gini (T) is defined as,

$$\text{gini}(T) = 1 - \sum_j^n (P_j)^2 \qquad ...(6.5.1)$$

In the above equation P_j represents the relative frequency of class j in T.

After splitting T into two subsets T_1 and T_2 with sizes N_1 and N_2, gini index of split data is,

$$\text{gini}_{\text{split}}(T) = \frac{N_1}{N}\text{gini}(T_1) + \frac{N_2}{N}\text{gini}(T_2) \qquad ... (6.5.2)$$

The attribute with smallest gini $_{\text{split}}$ (T) is selected to split the node.

2. Information Gain (ID3)

- In this method all attributes are assumed to be categorical. The method can be modified for continuous valued attributes. Here we select the attribute with highest information gain.
- Assume there are 2 classes P and N. Let the set of records S contain p records of class P and n records of class N.
- The amount of information required to decide if a random record in S belongs to P or N is defined as,

$$I(p, n) = -\left(\frac{p}{p+n}\right)\log_2\left(\frac{p}{p+n}\right) - \left(\frac{n}{p+n}\right)\log_2\left(\frac{n}{p+n}\right) \qquad ... (6.5.3)$$

- Assume that using attribute A, a set S will be partitioned in to sets $\{s_1, s_2, ...s_k\}$
- If S_i has p_i records of P and n_i records of N, the entropy or the expected information required to classify objects in all subtrees S_i is,

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i) \qquad ... (6.5.4)$$

- **Entropy (E) :** Expected amount of information (in bits) needed to assign a class to a randomly drawn object in S under the optimal shortest length code.
- **Gain (A) :** Measures reduction in entropy achieved because of split. Choose split that achieves most reduction (maximum Gain).

$$\text{Gain}(A) = I(p, n) - E(A) \qquad ...(6.5.5)$$

4. Avoid Overfitting in classification (Tree pruning)

The generated tree may overfit the training data.

- If there are too many branches then some may reflect anomalies due to noise or outliers.
- Overfitting result in poor accuracy for unseen samples.

There are two approaches to avoid overfitting, prune the tree so that it is not too specific.

- **Prepruning (prune while building tree)**

  Stop tree construction early do not divide a node if this would result in the goodness measure falling below threshold.

- **Postpruning (prune after building tree)**

  Fully constructed tree gets a sequence of progressively pruned trees.

5. Strengths of Decision Tree Method

- Able to generate understandable rules
- Performs classification without requiring much computation.
- Able to handle both continuous and categorical variables.
- Decision tree clearly indicates which fields are most important for prediction or classification.

Module 6

6. Weakness of Decision Tree Method

- Not suitable for prediction of continuous attribute
- Perform poorly with many class and small data
- Computationally expensive to train.

6.5.2 Constructing Decision Tree

- The ID3 algorithm starts with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the entropy (or information gain) of that attribute. Algorithm next selects the attribute which has the smallest entropy (or largest information gain) value.
- The set S is then divided by the chosen attribute (e.g. Income is less than 20 K , Income is between 20 K and 40 K, Income is greater than 40 K) to produce subsets of the data.
- The algorithm is recursively called for each subset, considering the attributes which are not selected before.
- The stopping criteria for recursion can be one of these situations :
  - When all records in the subset belongs to the same class (+ or –), then the node is converted into a leaf node and labelled with the class of the records.
  - When we have selected all the attributes, but the records still do not belong to the same class (some are + and some are –), then the node is converted into a leaf node and labelled with the most frequent class of the records in the subset
  - When there are no records in the subset, this is due to the non coverage of a specific attribute value for the record in the parent set, for example if there was no record with income = 40 K. Then a leaf node is generated and labelled with the most frequent class of the record in the parent set.
- Decision tree is generated with each non-terminal node representing the selected attribute on which the data was split, and terminal nodes representing the class label of the final subset of this branch.

Summary

Entropy of each and every attribute is calculated using the data set

- Divide the set S into subsets using the attribute for which the resulting entropy (after splitting) is minimum (or, equivalently, information gain is maximum)
- Make a decision tree node containing that attribute
- Recurse on subsets using remaining attributes.

Pseudocode

```
ID3 (Records, Target_Attribute, Attributes)
Generate a root node for the tree
If all records are positive, Return the single-node tree Root, with '+ 'label.
If all records are negative, Return the single-node tree Root, with '-' label.
If number of predicting attributes is empty, then return the single node tree Root, and label with most frequent value of the target attribute in the records.
Otherwise Begin
A ← The Attribute that best classifies records.
Decision Tree attribute for Root 'A'.
For each possible value, v_i, of A,
Add a new tree branch below Root, corresponding to the test A = v_i.
Let Record (v_i) be the subset of records that have the value v_i for A
If Record (v_i) is empty
Then below this new branch add a leaf node and label with most frequent target value in the records
Else below this new branch add the sub tree ID3 (Records (v_i), Target_Attribute,
Attributes – {A})
End
Return Root
```

NOTES

6.5.3 Example of Classification Tree using ID3

Ex. 6.5.1 : Suppose we want ID3 to evaluate car database as whether the car is acceptable or not. The target classification is "Should we accept car?" which can be acceptable or unacceptable.

Buying_Price	Maintenance_Price	Lug_Boot	Safety	Evaluation?
High	High	Small	High	Unacceptable
High	High	Small	Low	Unacceptable
Medium	High	Small	High	Acceptable
Low	Medium	Small	High	Acceptable
Low	Low	Big	High	Acceptable
Low	Low	Big	Low	Unacceptable
Medium	Low	Big	Low	Acceptable
High	Medium	Small	High	Unacceptable
High	Low	Big	High	Acceptable
Low	Medium	Big	High	Acceptable
High	Medium	Big	Low	Acceptable
Medium	Medium	Small	Low	Acceptable
Medium	High	Big	High	Acceptable
Low	Medium	Small	Low	Unacceptable

Soln. :

Class P : Evaluation = "Acceptable"

Class N: Evaluation = "Unacceptable"

Total records = 14

Number of records with Acceptable = 9 and Unacceptable = 5

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

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

Step 1

1. Compute entropy for Buying_Price

For Buying_Price = High

$$p_i = 2 \quad \text{and} \quad n_i = 3$$

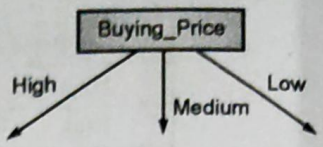
$$I(p_i, n_i) = I(2, 3) = \left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right)-\left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) = 0.971$$

Similarly we will calculate $I(p_i, n_i)$ for Medium and Low.

Buying_Price	p_i	n_i	$I(p_i, n_i)$
High	2	3	0.971
Medium	4	0	0
Low	3	2	0.971

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

$$E(\text{Buying\_Price}) = \left(\frac{5}{14}\right) I(2,3) + \left(\frac{4}{14}\right) I(4,0) + \left(\frac{5}{14}\right) I(3,2) = 0.694$$



$$\text{Gain}(S, \text{Buying\_Price}) = I(p, n) - E(\text{Buying\_Price}) = 0.940 - 0.694 = 0.246$$

Similarly, Gain (S, Maintenance_ Price) = 0.029, Gain (S, Lug_Boot) = 0.151, Gain (S, Safety) = 0.048

Since Buying_Price is the highest we select Buying_Price as the root node.

Step 2

As attribute Buying_Price at root, we have to decide on remaining tree attribute for High branch.

Buying Price	Maintenance_ Price	Lug_Boot	Safety	Evaluation?
High	High	Small	High	Unacceptable
High	High	Small	Low	Unacceptable
High	Medium	Small	High	Unacceptable
High	Low	Big	High	Acceptable
High	Medium	Big	Low	Acceptable

No. of records with Acceptable = 2 and Unacceptable = 3

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

$$I(2, 3) = -\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) = 0.971$$

1. Compute entropy for Maintenance_ Price

Maintenance_ Price	p_i	n_i	$I(P_i, n_i)$
High	0	2	0
Medium	1	1	1
Low	1	0	0

$$E(\text{Maintenance\_ Price}) = \left(\frac{2}{5}\right) I(0,2) + \left(\frac{2}{5}\right) I(1,1) + \left(\frac{1}{5}\right) I(1,0) = 0.4$$

$$\text{Gain}(S_{\text{High}}, \text{Maintenance\_ Price}) = I(p, n) - E(\text{Maintenance\_ Price}) = 0.971 - 0.4 = 0.571$$

2. Compute entropy for Lug_Boot

$P_i = 0$ and $n_i = 3$

$I(P_i, n_i) = I(0, 3) = 0$

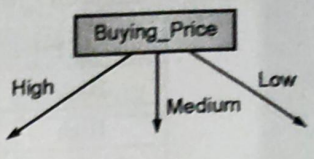
Lug_Boot	p_i	n_i	$I(P_i, n_i)$
Small	0	3	0
Big	2	0	0

$$E(\text{Lug\_Boot}) = \left(\frac{3}{5}\right) I(0, 3) + \left(\frac{2}{5}\right) I(2, 0) = 0$$

$$\text{Gain}(S_{\text{High}}, \text{Lug\_Boot}) = I(p, n) - E(\text{Lug\_Boot}) = 0.971 - 0 = 0.971$$

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

$$E(\text{Buying\_Price}) = \left(\frac{5}{14}\right) I(2,3) + \left(\frac{4}{14}\right) I(4,0) + \left(\frac{5}{14}\right) I(3,2) = 0.694$$



$$\text{Gain}(S, \text{Buying\_Price}) = I(p, n) - E(\text{Buying\_Price}) = 0.940 - 0.694 = 0.246$$

Similarly, Gain (S, Maintenance_ Price) = 0.029, Gain (S, Lug_Boot) = 0.151, Gain (S, Safety) = 0.048

Since Buying_Price is the highest we select Buying_Price as the root node.

▶ **Step 2**

As attribute Buying_Price at root, we have to decide on remaining tree attribute for High branch.

Buying Price	Maintenance_ Price	Lug_Boot	Safety	Evaluation?
High	High	Small	High	Unacceptable
High	High	Small	Low	Unacceptable
High	Medium	Small	High	Unacceptable
High	Low	Big	High	Acceptable
High	Medium	Big	Low	Acceptable

No. of records with Acceptable = 2 and Unacceptable = 3

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

$$I(2, 3) = -\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) = 0.971$$

1. Compute entropy for Maintenance_ Price

Maintenance_ Price	p_i	n_i	$I(P_i, n_i)$
High	0	2	0
Medium	1	1	1
Low	1	0	0

$$E(\text{Maintenance\_ Price}) = \left(\frac{2}{5}\right) I(0,2) + \left(\frac{2}{5}\right) I(1,1) + \left(\frac{1}{5}\right) I(1,0) = 0.4$$

$$\text{Gain}(S_{\text{High}}, \text{Maintenance\_ Price}) = I(p, n) - E(\text{Maintenance\_ Price}) = 0.971 - 0.4 = 0.571$$

2. Compute entropy for Lug_Boot

$P_i = 0$ and $n_i = 3$

$I(P_i, n_i) = I(0, 3) = 0$

Lug_Boot	p_i	n_i	$I(P_i, n_i)$
Small	0	3	0
Big	2	0	0

$$E(\text{Lug\_Boot}) = \left(\frac{3}{5}\right) I(0, 3) + \left(\frac{2}{5}\right) I(2, 0) = 0$$

$$\text{Gain}(S_{\text{High}}, \text{Lug\_Boot}) = I(p, n) - E(\text{Lug\_Boot}) = 0.971 - 0 = 0.971$$

3. Compute entropy for Safety

$p_i = 1$ and $n_i = 2$

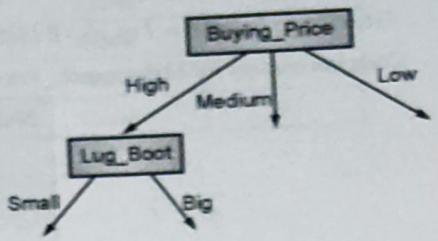
$I(p_i, n_i) = I(1, 2) = 0.918$

Safety	p_i	n_i	$I(P_i, n_i)$
High	1	2	0.918
Low	1	1	1

$$E(\text{Safety}) = \left(\frac{3}{5}\right) I(1, 2) + \left(\frac{2}{5}\right) I(1, 1) = 0.951$$

$$\text{Gain}(S_{\text{High}}, \text{Safety}) = I(p, n) - E(\text{Safety}) = 0.971 - 0.951 = 0.02$$

Since Lug_Boot is the highest we select Lug_Boot as a next node below High branch.

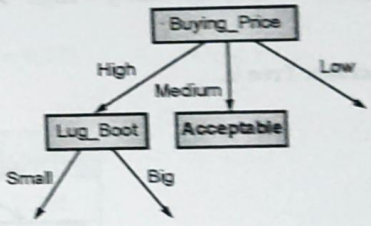


▶ **Step 3**

Consider now only Maintenance_ Price and Safety for Buying_Price = Medium

Buying_Price	Maintenance_ Price	Lug_Boot	Safety	Evaluation?
Medium	High	Small	High	Acceptable
Medium	Low	Big	Low	Acceptable
Medium	Medium	Small	Low	Acceptable
Medium	High	Big	High	Acceptable

Since for any combination of values of Maintenance_ Price and Safety, Evaluation? value is Acceptable, so we can directly write down the answer as Acceptable.



▶ **Step 4**

Consider now only Maintenance_ Price and Safety for Buying_Price = Low

Buying_Price	Maintenance_ Price	Lug_Boot	Safety	Evaluation?
Low	Medium	Small	High	Acceptable
Low	Low	Big	High	Acceptable
Low	Low	Big	Low	Unacceptable
Low	Medium	Big	High	Acceptable
Low	Medium	Small	Low	Unacceptable

$P_i = 3$ and $n_i = 2$

$I(P_i, n_i) = I(3,2) = 0.970$

1. Compute entropy for Safety

Safety	p_i	n_i	$I(P_i, n_i)$
High	3	0	0
Low	0	2	0

$$E(\text{Safety}) = \left(\frac{3}{5}\right) I(3, 0) + \left(\frac{2}{5}\right) I(0, 2) = 0$$

$$\text{Gain}(S_{\text{Low}}, \text{Safety}) = I(p, n) - E(\text{Safety}) = 0.970 - 0 = 0.970$$

2. Compute entropy for Maintenance_ Price

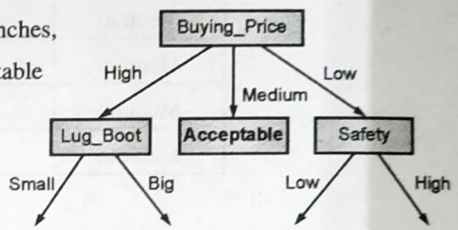
Maintenance_ Price	p_i	n_i	$I(P_i, n_i)$
High	0	0	0
Medium	2	1	0.918
Low	1	1	1

$$E(\text{Maintenance\_ Price}) = \left(\frac{0}{5}\right) I(0, 0) + \left(\frac{3}{5}\right) I(2, 1) + \left(\frac{2}{5}\right) I(1, 1) = 0.951$$

$$\text{Gain}(S_{\text{Low}}, \text{Maintenance\_ Price}) = I(p, n) - E(\text{Maintenance\_ Price}) = 0.970 - 0.951 = 0.019$$

Since, Safety is the highest we select Safety below Low branch.

- Now we will check the value of 'Evaluation?' from the database, for all branches,
  - Buying_Price = High and Lug_Boot = Small –> Evaluation? = Unacceptable
  - Buying_Price = High and Lug_Boot = Big –> Evaluation? = Acceptable
  - Buying_Price = Low and Safety = Low –> Evaluation? = Unacceptable
  - Buying_Price = Low and Safety = High –> Evaluation? = Acceptable



- **Final Decision Tree is**

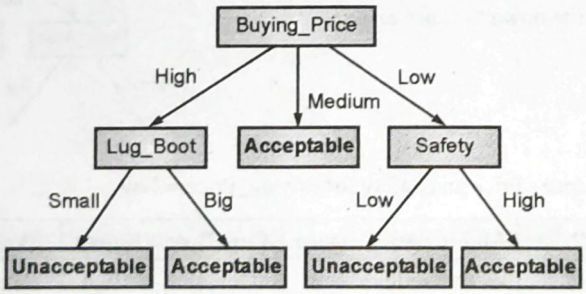


Fig. Ex. 6.5.1

Ex. 6.5.2 : Suppose we want ID3 to decide whether the loan is to be sanctioned or not. The target classification is "Should we sanction loan?" which can be yes or no.

Customer no	Spending_ Habit	Collateral	Income	Credit_Score	Sanction?
1	High	None	Low	Bad	No
2	High	None	Medium	Unknown	No
3	Low	None	Medium	Unknown	No
4	Low	None	Low	Unknown	No

Customer no	Spending_ Habit	Collateral	Income	Credit_Score	Sanction?
5	Low	None	High	Unknown	Yes
6	Low	Sufficient	High	Unknown	Yes
7	Low	None	Medium	Bad	No
8	Low	Sufficient	High	Bad	No
9	Low	None	High	Good	Yes
10	High	Sufficient	High	Good	Yes
11	High	None	Low	Good	No
12	High	None	Medium	Good	No

☑ **Soln. :**

Class P : Sanction = "Yes"

Class N : Sanction = "No"

Total records = 12

No. of records with Yes = 4 and No = 8

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

$$I(4, 8) = -\left(\frac{4}{12}\right)\log_2\left(\frac{4}{12}\right)-\left(\frac{8}{12}\right)-\log_2\left(\frac{8}{12}\right) = 0.922$$

▶ **Step 1**

1. Compute entropy for Spending_Habit

For Spending_Habit = High

$p_i = 1$ and $n_i = 4$

$$I(p_i, n_i) = I(1, 4) = -\left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right)-\left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right) = 0.721$$

Similarly, we will calculate $I(p_i, n_i)$ for Low.

Spending_Habit	p_i	n_i	$I(p_i, n_i)$
High	1	4	0.721
Low	3	4	0.985

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

$$E(\text{Spending\_Habit}) = \left(\frac{5}{12}\right)\times 0721 + \left(\frac{7}{12}\right)\times 0.985 = 0.874$$

Gain(S, Spending_Habit) = I (p, n) – E (Spending_Habit) = 0.922 – 0.874 = 0.048

2. Compute entropy for Collateral

Collateral	p_i	n_i	$I(p_i, n_i)$
None	2	7	0.77
Sufficient	2	1	0.918

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

$$E(\text{Collateral}) = \left(\frac{9}{12}\right) \times 0.77 + \left(\frac{3}{12}\right) \times 0.918 = 0.806$$

$$\text{Gain(S, Collateral)} = I(p, n) - E(\text{Collateral}) = 0.922 - 0.806 = 0.116$$

3. Compute entropy for Income

Income	p_i	n_i	$I(p_i, n_i)$
Low	0	3	0
Medium	0	4	0
High	4	1	0.721

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

$$E(\text{Income}) = \left(\frac{9}{12}\right) \times 0 + \left(\frac{4}{12}\right) \times 0 + \left(\frac{5}{12}\right) \times 0.721 = 0.3$$

$$\text{Gain(S, Income)} = I(p, n) - E(\text{Income}) = 0.922 - 0.3 = 0.622$$

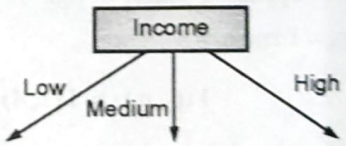
4. Compute entropy for Credit_Score

Credit_Score	p_i	n_i	$I(p_i, n_i)$
Bad	0	3	0
Unknown	2	3	0.97
Good	2	2	1

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

$$E(\text{Credit\_Score}) = \left(\frac{3}{12}\right) \times 0 + \left(\frac{5}{12}\right) \times 0.97 + \left(\frac{4}{12}\right) \times 1 = 0.734$$

$$\text{Gain(S, Credit\_Score)} = I(p, n) - E(\text{Credit\_Score}) = 0.922 - 0.734 = 0.188$$



Since Income is the highest we select Income as the root node.

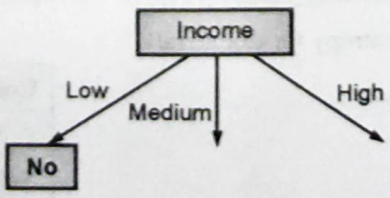
▶ **Step 2**

As attribute Income at root, we have to decide on remaining tree attribute for Low branch.

Consider only Spending_Habit, Collateral and Credit_Score for Income = Low

Customer no	Spending_ Habit	Collateral	Income	Credit_Score	Sanction?
1	High	None	Low	Bad	No
4	Low	None	Low	Unknown	No
11	High	None	Low	Good	No

Since for any combination of values of Spending_Habit, Collateral and Credit_Score, Sanction? value is No for Income = Low, so we can directly write down the answer as No

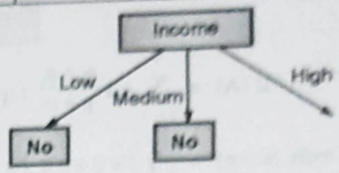


Step 3

Consider only Spending_Habit, Collateral and Credit_Score Income = Medium

Customer no	Spending_ Habit	Collateral	Income	Credit_Score	Sanction?
2	High	None	Medium	Unknown	No
3	Low	None	Medium	Unknown	No
7	Low	None	Medium	Bad	No
12	High	None	Medium	Good	No

Since for any combination of values of Spending_Habit, Collateral and Credit_Score, Sanction? value is No for Income = Medium, so we can directly write down the answer as No



Step 4

Consider only Spending_Habit, Collateral and Credit_Score Income = High

Customer no	Spending_ Habit	Collateral	Income	Credit_Score	Sanction?
5	Low	None	High	Unknown	Yes
6	Low	Sufficient	High	Unknown	Yes
8	Low	Sufficient	High	Bad	No
9	Low	None	High	Good	Yes
10	High	Sufficient	High	Good	Yes

No. of records with Yes = 4 and No = 1

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

$$I(2, 3) = -\left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right) - \left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right) = 0.721$$

1. Compute entropy for Spending_Habit

Spending_Habit	p_i	n_i	$I(p_i, n_i)$
Low	3	1	0.811
High	1	0	0

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

$$E(\text{Spending\_Habit}) = \left(\frac{4}{5}\right) \times 0811 + \left(\frac{1}{5}\right) \times 0 = 0.648$$

Gain (S_{High}, Spending_Habit) = I (p, n) – E (Spending_Habit) = 0.721 – 0.648 = 0.073

2. Compute entropy for Collateral

Collateral	p_i	n_i	$I(p_i, n_i)$
None	2	0	0
Sufficient	2	1	0.918

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

$$E(\text{Collateral}) = \left(\frac{2}{5}\right) \times 0 + \left(\frac{3}{5}\right) \times 0.918 = 0.55$$

$$\text{Gain}(S_{High}, \text{Collateral}) = I(p, n) - E(\text{Collateral}) = 0.721 - 0.55 = 0.171$$

3. Compute entropy for Credit_Score

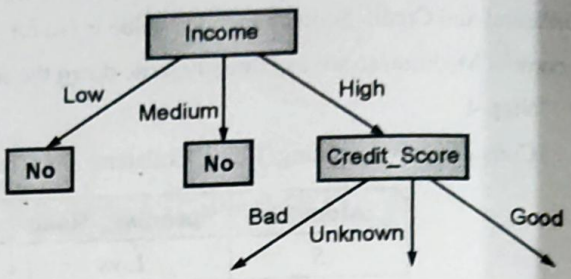
Credit_Score	p_i	n_i	$I(p_i, n_i)$
Unknown	1	0	0
Bad	0	1	0
Good	2	0	0

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

$$E(\text{Credit\_Score}) = \left(\frac{1}{5}\right) \times 0 + \left(\frac{1}{5}\right) \times 0 + \left(\frac{2}{5}\right) \times 0 = 0$$

$$\text{Gain}(S_{High}, \text{Credit\_Score}) = I(p, n) - E(\text{Credit\_Score})$$

$$= 0.721 - 0 = 0.721$$



- Since Credit_Score is the highest we select Credit_Score as a next node below High branch.
- Now we will check the value of 'Sanction?' from the database, for all branches,
  - Income = High and Credit_Score = Bad -> Sanction? = No
  - Income = High and Credit_Score = Unknown -> Sanction? = Yes
  - Income = High and Credit_Score = Good -> Sanction? = Yes
- Final Decision Tree is

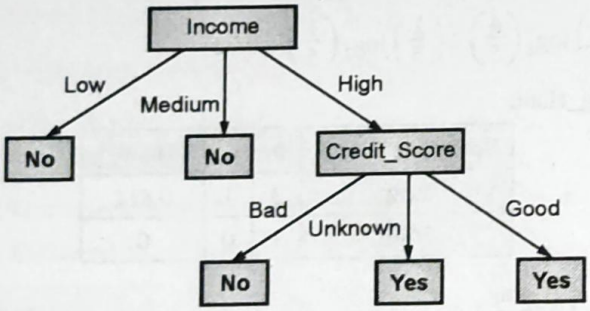


Fig. Ex. 6.5.2

Ex. 6.5.3 : Suppose we want ID3 to decide whether the car will be stolen or not. The target classification is "car is stolen?" which can be Yes or No.

Car no	Colour	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No

Car no	Colour	Type	Origin	Stolen?
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

☑ **Soln. :**

Class P: Stolen = "Yes"

Class N: Stolen = "No"

Total records = 10

No. of records with Yes = 5 and No = 5

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

$$I(5, 5) = -\left(\frac{5}{10}\right)\log_2\left(\frac{5}{10}\right) - \left(\frac{5}{10}\right)\log_2\left(\frac{5}{10}\right) = 1$$

▶ **Step 1**

1. Compute entropy for Colour

Colour	p_i	n_i	$I(p_i, n_i)$
Red	3	2	0.971
Yellow	2	3	0.971

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

$$E(\text{Colour}) = \left(\frac{5}{10}\right) \times 0.971 + \left(\frac{5}{10}\right) \times 0.971 = 0.971$$

$$\text{Gain}(S, \text{Colour}) = I(p, n) - E(\text{Colour}) = 1 - 0.971 = 0.029$$

2. Compute entropy for Type

Type	p_i	n_i	$I(p_i, n_i)$
Sports	4	2	0.923
SUV	1	3	0.811

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

$$E(\text{Type}) = \left(\frac{6}{10}\right) \times 0.923 + \left(\frac{4}{10}\right) \times 0.811 = 0.878$$

$$\text{Gain}(S, \text{Type}) = I(p, n) - E(\text{Type}) = 1 - 0.878 = 0.1218$$

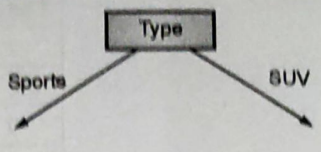
3. Compute entropy for Origin

Origin	p_i	n_i	$I(p_i, n_i)$
Domestic	2	3	0.971
Imported	3	2	0.971

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

$$E(\text{Origin}) = \left(\frac{5}{10}\right) \times 0.971 + \left(\frac{5}{10}\right) \times 0.971 = 0.971$$

$$\text{Gain}(S, \text{Origin}) = I(p, n) - E(\text{Origin}) = 1 - 0.971 = 0.029$$



Since Type is the highest we select Type as the root node.

▶ **Step 2**

As attribute Type at root, we have to decide on remaining tree attribute for Sports branch.

Consider only Colour and Origin for Type = Sports

Car no	Colour	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
10	Red	Sports	Imported	Yes

No. Of records with Yes = 4 and No = 2

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

$$I(4, 2) = -\left(\frac{4}{6}\right) \log_2 \left(\frac{4}{6}\right) - \left(\frac{2}{6}\right) \log_2 \left(\frac{2}{6}\right) = 0.923$$

1. Compute entropy for Colour

Colour	p_i	n_i	$I(p_i, n_i)$
Red	3	1	0.811
Yellow	1	1	1

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

$$E(\text{Colour}) = \left(\frac{4}{6}\right) \times 0.811 + \left(\frac{2}{6}\right) \times 1 = 0.873$$

$$\text{Gain}(S_{\text{Sports}}, \text{Colour}) = I(p, n) - E(\text{Colour}) = 0.923 - 0.873 = 0.05$$

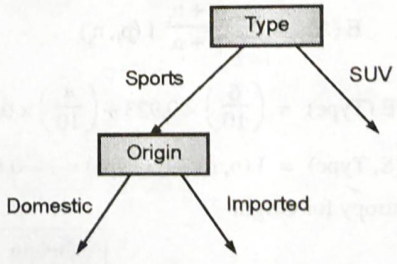
2. Compute entropy for Origin

Origin	p_i	n_i	$I(p_i, n_i)$
Domestic	2	2	1
Imported	2	0	0

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

$$E(\text{Origin}) = \left(\frac{4}{6}\right) \times 1 + \left(\frac{2}{6}\right) \times 0 = 0.666$$

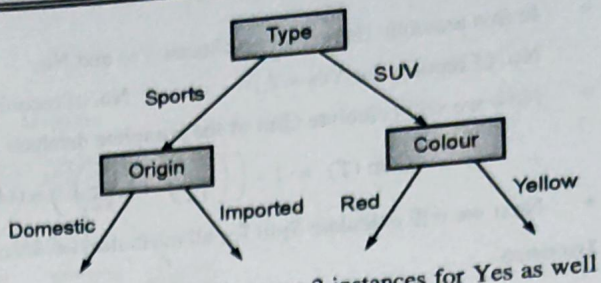
$$\text{Gain}(S_{\text{Sports}}, \text{Origin}) = I(p, n) - E(\text{Origin}) = 0.923 - 0.666 = 0.257$$



Since Origin is the highest we select as a next node below Sports branch.

Step 3

- As attribute Type and Origin is already chosen, we have to decide on only remaining Colour attribute for SUV branch.



- Now we will check the value of 'Stolen?' from the database, for all branches,
  - For, Type = Sports and Origin = Domestic, Stolen? = Yes as well as No
- So for this type of case we have to select the most common class. In this example there are 2 instances for Yes as well as No, so we can select any one. Let's we select No.
  - For, Type = Sports and Origin = Imported, Stolen? = Yes
  - For, Type = SUV and Colour = Red, Stolen? = No
  - For, Type = SUV and Colour = Yellow, Stolen? = Yes as well as No
- So for this type of case we have to select the most common class. In this example there are 2 instances for No and 1 instance of Yes, so we will select No.
- Final Decision Tree is

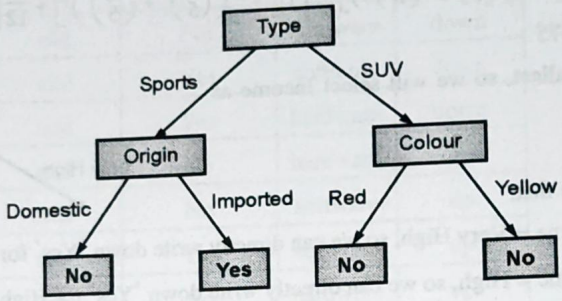


Fig. Ex. 6.5.3

6.5.4 Example of Classification Tree using Gini Index

Ex. 6.5.4 : Create a decision tree using Gini Index to classify following dataset.

Sr. No.	Income	Age	Own Car
1	Very High	Young	Yes
2	High	Medium	Yes
3	Low	Young	No
4	High	Medium	Yes
5	Very High	Medium	Yes
6	Medium	Young	Yes
7	High	Old	Yes
8	Medium	Medium	No
9	Low	Medium	No
10	Low	Old	No
11	High	Young	Yes
12	Medium	Old	No

☑ **Soln. :**

- In this example there are two classes Yes and No.

  No. of records for Yes = 7 ; No. of records for No = 5 ; Total No. of records = 12

- Now we will calculate Gini of the complete database as,

$$\text{Gini (T)} = 1 - \left(\left(\frac{7}{12}\right)^2 + \left(\frac{5}{12}\right)^2\right) = 0.48$$

- Next we will calculate Split for all attributes, i.e. Income and Age.

Income

$$\text{Split} = \frac{2}{12}\text{gini (Very High)} + \frac{4}{12}\text{gini (High)} + \frac{3}{12}\text{gini (Low)} + \frac{3}{12}\text{gini (Medium)}$$

$$= \frac{2}{12}\left[1 - \left(\left(\frac{2}{2}\right)^2 + \left(\frac{0}{2}\right)^2\right)\right] + \frac{4}{12}\left[1 - \left(\left(\frac{4}{4}\right)^2 + \left(\frac{0}{4}\right)^2\right)\right] + \frac{3}{12}\left[1 - \left(\left(\frac{0}{3}\right)^2 + \left(\frac{3}{3}\right)^2\right)\right] + \frac{3}{12}\left[1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right)\right]$$

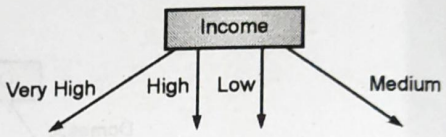
$$= 0.1125$$

$$\text{Age Split} = \frac{4}{12}\text{gini (Yong)} + \frac{5}{12}\text{gini (Medium)} + \frac{3}{12}\text{gini (Old)}$$

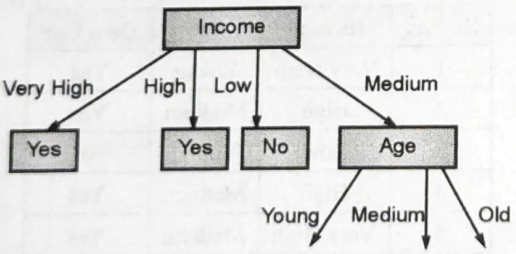
$$= \frac{4}{12}\left[1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right)\right] + \frac{5}{12}\left[1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right)\right] + \frac{3}{12}\left[1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right)\right]$$

$$= 0.4375$$

- Split value of Income is smallest, so we will select Income as root node.



- From the database we can see that,
  - Own Car = Yes for Income = Very High, so we can directly write down 'Yes' for Very High branch.
  - Own Car = Yes for Income = High, so we can directly write down 'Yes' for High branch.
  - Own Car = No for Income = Low, so we can directly write down 'No' for Low branch.
- Since Income is taken as root node, now we have to decide on the Age attribute, so we will take Age as next node below Medium branch.



- From the database we can see that,
  - Own Car = Yes for Income = Medium and Age = Young, so we can directly write down 'Yes' for Young branch.
  - Own Car = No for Income = Medium and Age = Medium, so we can directly write down 'No' for medium branch.
  - Own Car = No for Income = Medium and Age = Old, so we can directly write down 'No' for Old branch.

- Final Decision Tree is,

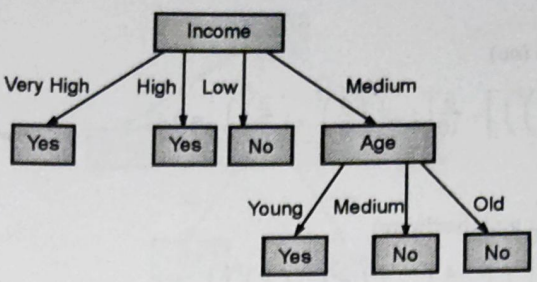


Fig. Ex. 6.5.4

Ex. 6.5.5 : Stock market involving only Discrete ranges has profit as categorical value {up, down}.Use Gini index method to draw classification tree.

Age	Competition	Type	Profit
old	Yes	software	down
old	No	software	down
old	No	hardware	down
mid	Yes	software	down
mid	Yes	hardware	down
mid	No	hardware	up
mid	No	software	up
new	Yes	software	up
new	No	hardware	up
new	no	software	up

☑ **Soln. :**

- In this example there are two classes down and up.

  No. of records for down = 5

  No. of records for up = 5

  Total No. of records = 10

- Now we will calculate Gini of the complete database as,

$$\text{Gini (T)} = \left[1-\left(\left(\frac{5}{10}\right)^2+\left(\frac{5}{10}\right)^2\right)\right]=0.5$$

- Next we will calculate Split for all attributes, i.e. Age, Competition and Type.

Age

$$\text{Split} = \frac{3}{10}\text{gini (old)} + \frac{4}{10}\text{gini (mid)}\frac{3}{10}\text{ gini (new)}$$

$$= \frac{3}{10}\left[1-\left(\left(\frac{0}{3}\right)^2+\left(\frac{3}{3}\right)^2\right)\right]+\frac{4}{10}\left[1-\left(\left(\frac{2}{4}\right)^2+\left(\frac{2}{4}\right)^2\right)\right]+\frac{3}{10}\left[1-\left(\left(\frac{3}{3}\right)^2+\left(\frac{0}{3}\right)^2\right)\right]=0.2$$

Competition

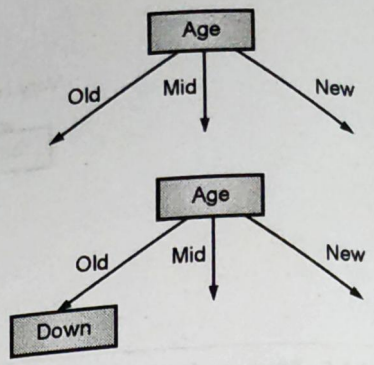
$$\text{Split} = \frac{4}{10}\text{gini (yes)} + \frac{6}{10}\text{gini (no)}$$

$$= \frac{4}{10}\left[1-\left(\left(\frac{1}{4}\right)^2+\left(\frac{3}{4}\right)^2\right)\right]+\frac{6}{10}\left[1-\left(\left(\frac{4}{6}\right)^2+\left(\frac{2}{6}\right)^2\right)\right]=0.42$$

Type

$$\text{Split} = \frac{6}{10}\text{gini (software)} + \frac{4}{10}\text{gini (hardware)}$$

$$= \frac{6}{10}\left[1-\left(\left(\frac{3}{6}\right)^2+\left(\frac{3}{6}\right)^2\right)\right]+\frac{4}{10}\left[1-\left(\left(\frac{2}{4}\right)^2+\left(\frac{2}{4}\right)^2\right)\right]=0.5$$



- Split value of Age is smallest, so we will select Age as root node.
- From the database we can see that, Profit = Down for Age = Old, so we can directly write down 'Down' for Old branch node.
- Next we will check for Age = mid

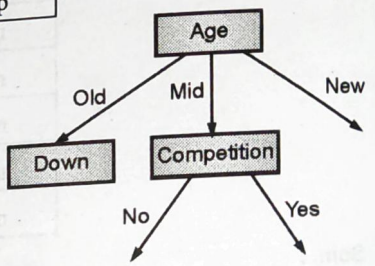
Age	Competition	Type	Profit
mid	Yes	software	Down
mid	Yes	hardware	Down
mid	No	hardware	Up
mid	No	software	Up

- Now we will calculate Split for only Competition and Type attributes.

Competition

$$\text{Split} = \frac{2}{4}\text{gini (yes)} + \frac{2}{4}\text{gini (no)}$$

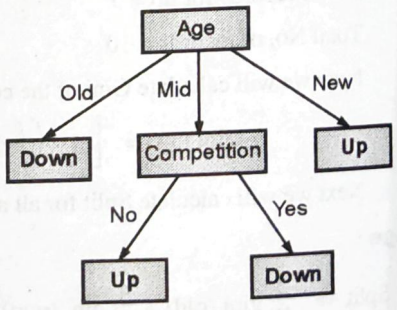
$$= \frac{2}{4}\left[1-\left(\left(\frac{0}{2}\right)^2+\left(\frac{2}{2}\right)^2\right)\right]+\frac{2}{4}\left[1-\left(\left(\frac{2}{2}\right)^2+\left(\frac{0}{2}\right)^2\right)\right]=0$$



Type

$$\text{Split} = \frac{2}{4}\text{gini (software)} + \frac{2}{4}\text{gini (hardware)}$$

$$= \frac{2}{4}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]+\frac{2}{4}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]=0.5$$



- Split value of Competition is smallest, so we will select Competition as next node below mid branch.
- From the database we can see that, Profit = Up for Age = New, so we can directly write down 'Up' for New branch node.
- Now we will check the value of 'Profit' from the database, for all branches,
  - Age = Mid and Competition = No -> Profit = Up
  - Age = Mid and Competition = Yes -> Profit = Down

- Final Decision Tree is,

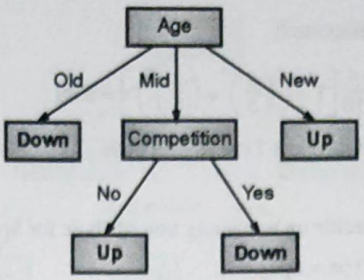


Fig. Ex. 6.5.5

Ex. 6.5.6 : Suppose we want Gini index to decide whether the car will be stolen or not. The target classification is "car is stolen?" which can be Yes or No.

Car no	Colour	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

☑ **Soln. :**

- In this example there are two classes Yes and No.

  No. of records for Yes = 5 ; No. of records for No = 5 ; Total No. of records = 10

- Now we will calculate Gini of the complete database as,

$$\text{Gini (T)} = 1 - \left(1 - \left(\left(\frac{5}{10}\right)^2 + \left(\frac{5}{10}\right)^2\right)\right) = 0.5$$

- Next we will calculate Split for all attributes, i.e. Colour, Type and Origin.

☞ **Colour**

$$\textbf{Split} = 5/10 \text{ gini (Red)} + 5/10 \text{ gini (Yellow)}$$

$$= \frac{5}{10}\left[1 - \left(\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right)\right] + \frac{5}{10}\left[1 - \left(\left(\frac{2}{5}\right)^2 + \left(\frac{3}{5}\right)^2\right)\right] = \mathbf{0.48}$$

☞ **Type**

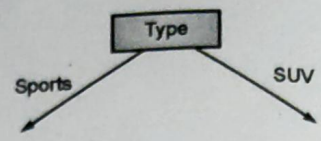
$$\text{Split} = \frac{6}{10}\text{ gini (Sports)} + \frac{4}{10}\text{ gini (SUV)}$$

$$= \frac{6}{10}\left[1 - \left(\left(\frac{4}{6}\right)^2 + \left(\frac{2}{6}\right)^2\right)\right] + \frac{4}{10}\left[1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right)\right] = \mathbf{0.42}$$

☞ **Origin**

$$Split = \frac{5}{10}\text{ gini (Domestic)} + \frac{5}{10}\text{ gini (imported)}$$

$$= \frac{5}{10}\left[1-\left(\left(\frac{2}{5}\right)^2+\left(\frac{3}{5}\right)^2\right)\right]+\frac{5}{10}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right] = 0.48$$



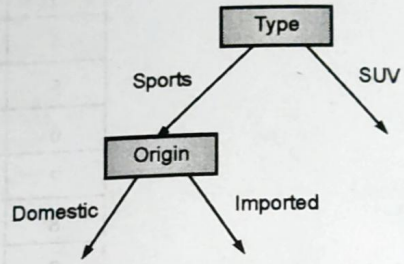
- Split value of Type is smallest, so we will select Type as root node.
- Next we will check for Type = Sports
- As attribute Type at root, we have to decide on remaining tree attribute for Sports branch.
- Consider only Colour and Origin for Type = Sports

Car no	Colour	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
10	Red	Sports	Imported	Yes

☞ **Colour**

$$Split = \frac{4}{6}\text{gini (Red)} + \frac{2}{6}\text{gini (Yellow)}$$

$$= \frac{4}{6}\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]+\frac{2}{6}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right] = 0.417$$



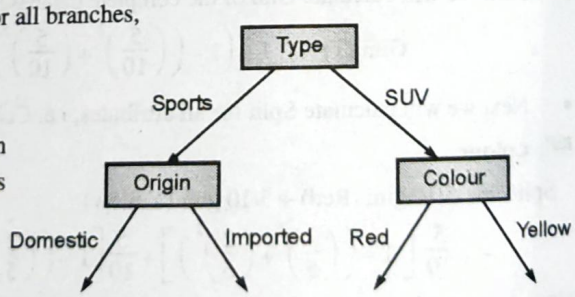
☞ **Origin>**

$$Split = \frac{4}{6}\text{gini (Domestic)} + \frac{4}{6}\text{gini (Imported)}$$

$$= \frac{4}{6}\left[1-\left(\left(\frac{2}{4}\right)^2+\left(\frac{2}{4}\right)^2\right)\right]+\frac{2}{6}\left[1-\left(\left(\frac{2}{2}\right)^2+\left(\frac{0}{2}\right)^2\right)\right] = 0.33$$

- Split value of Origin is smallest, so we will select Origin as next node.
- Next we will check for Type = SUV
- As attribute Type and Origin is already chosen, we have to decide on only remaining Colour attribute for SUV branch.
- Now we will check the value of 'Stolen?' from the database, for all branches,

  For, Type = Sports and Origin = Domestic, Stolen? = Yes as well as No
- So for this type of case we have to select the most common class. In this example there are 2 instances for Yes as well as No, so we can select any one. Let's we select No.
  - For, Type = Sports and Origin = Imported, Stolen? = Yes
  - For, Type = SUV and Colour = Red, Stolen? = No
  - For, Type = SUV and Colour = Yellow, Stolen? = Yes as well as No
- So for this type of case we have to select the most common class (Since all attributes are already considered). In this example there are 2 instances for No and 1 instance of Yes, so we will select No.



- Final Decision Tree is

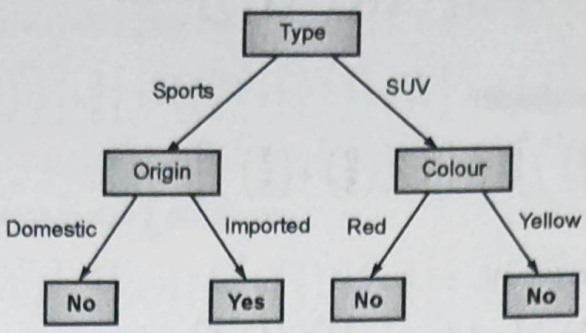


Fig. Ex. 6.5.6

UEx. 6.5.7 **MU - May 15, 12 Marks**

Create a decision tree for the attribute "class" using the respective values :

Eyecolour	Married	Sex	Hairlength	class
Brown	yes	Male	Long	Football
Blue	yes	Male	Short	Football
Brown	yes	Male	Long	Football
Brown	no	Female	Long	Netball
Brown	no	Female	Long	Netball
Blue	no	Male	Long	Football
Brown	no	Female	Long	Netball
Brown	no	Male	Short	Football
Brown	yes	Female	Short	Netball
Brown	no	Female	Long	Netball
Blue	no	Male	Long	Football
Blue	no	Male	Short	Football

☑ **Soln. :**

In this example there are two classes Football and Netball.

No. Of records for Football = 7 ; No. Of records for Netball = 5 ; Total No. Of records = 12

Now we will calculate Gini of the complete database as,

$$\text{Gini (T)} = 1 - \left(\left(\frac{7}{12}\right)^2 + \left(\frac{5}{12}\right)^2 \right) = 0.48$$

Next we will calculate Split for all attributes, i.e. Eyecolor, Married, Sex and Hairlength.

Eyecolor->

$$\text{Split} = \frac{8}{12}\text{gini (Brown)} + \frac{4}{12}\text{gini (Blue)}$$

$$= \frac{8}{12}\left[1 - \left(\left(\frac{3}{8}\right)^2 + \left(\frac{5}{8}\right)^2 \right) \right] + \frac{4}{12}\left[1 - \left(\left(\frac{4}{4}\right)^2 + \left(\frac{0}{4}\right)^2 \right) \right] = 0.31$$

Married->

$$\text{Split} = \frac{4}{12}\text{gini (yes)} + \frac{8}{12}\text{gini (no)}$$

$$= \frac{4}{12}\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]+\frac{8}{12}\left[1-\left(\left(\frac{4}{8}\right)^2+\left(\frac{4}{8}\right)^2\right)\right]=0.458$$

Sex->

$$\text{Split} = \frac{7}{12}\text{gini (male)} + \frac{5}{12}\text{gini (female)}$$

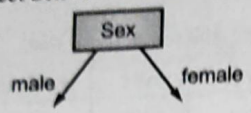
$$= \frac{7}{12}\left[1-\left(\left(\frac{7}{7}\right)^2+\left(\frac{0}{7}\right)^2\right)\right]+\frac{5}{12}\left[1-\left(\left(\frac{0}{5}\right)^2+\left(\frac{5}{5}\right)^2\right)\right]=0$$

Hairlength->

$$\text{Split} = \frac{8}{12}\text{gini (long)} + \frac{4}{12}\text{gini (short)}$$

$$= \frac{8}{12}\left[1-\left(\left(\frac{4}{8}\right)^2+\left(\frac{4}{8}\right)^2\right)\right]+\frac{4}{12}\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]=0.458$$

Split value of Sex is smallest, so we will select Sex as root node.



From the database we can see that,

class = Football for Sex = male, so we can directly write down 'Football' for male branch.

class = Netball for Sex = female, so we can directly write down 'Netball' for female branch.

Final decision tree is,

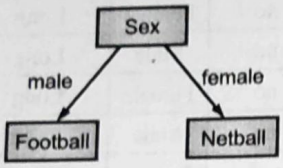


Fig. Ex. 6.5.7

UEx. 6.5.8 **MU - May 17, 10Marks**

For a Sunburn dataset given below, construct a decision tree

Name	Hair	Height	Weight	Location	Class
Swati	Blonde	Average	Light	No	Yes
Sunita	Blonde	Tall	Average	Yes	No
Anita	Brown	Short	Average	Yes	No
Lata	Blonde	Short	Average	No	Yes
Radha	Red	Average	Heavy	No	Yes
Maya	Brown	Tall	Heavy	No	No
Leena	Brown	Average	Heavy	No	No
Rina	Blonde	Short	Light	Yes	No

☑ **Soln. :**

We will calculate Split for all attributes, i.e. Hair, Height, Weight and Location.

Hair->

$$\text{Split} = \frac{4}{8}\text{gini (Blonde)} + \frac{3}{8}\text{gini (Brown)} + \frac{1}{8}\text{gini (Red)}$$

$$= \frac{4}{8}\left[1-\left(\left(\frac{2}{4}\right)^2+\left(\frac{2}{4}\right)^2\right)\right]+\frac{3}{8}\left[1-\left(\left(\frac{0}{3}\right)^2+\left(\frac{3}{3}\right)^2\right)\right]+\frac{1}{8}\left[1-\left(\left(\frac{1}{1}\right)^2+\left(\frac{0}{1}\right)^2\right)\right]=0.25$$

Height->

$$\text{Split} = \frac{3}{8}\text{gini (Average)} + \frac{2}{8}\text{gini (Tall)} + \frac{3}{8}\text{gini (Short)}$$

$$= \frac{3}{8}\left[1-\left(\left(\frac{2}{3}\right)^2+\left(\frac{1}{3}\right)^2\right)\right]+\frac{2}{8}\left[1-\left(\left(\frac{0}{2}\right)^2+\left(\frac{2}{2}\right)^2\right)\right]+\frac{3}{12}\left[1-\left(\left(\frac{1}{3}\right)^2+\left(\frac{2}{3}\right)^2\right)\right]=0.40$$

Weight->

$$\text{Split} = \frac{2}{8}\text{gini (Light)} + \frac{3}{8}\text{gini (Average)} + \frac{3}{8}\text{gini (Heavy)}$$

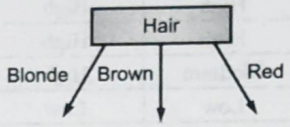
$$= \frac{2}{8}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]+\frac{3}{8}\left[1-\left(\left(\frac{1}{3}\right)^2+\left(\frac{2}{3}\right)^2\right)\right]+\frac{3}{8}\left[1-\left(\left(\frac{1}{3}\right)^2+\left(\frac{2}{3}\right)^2\right)\right]=0.525$$

Location->

$$\text{Split} = \frac{5}{8}\text{gini (No)} + \frac{3}{8}\text{gini (Yes)}$$

$$= \frac{5}{8}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]+\frac{3}{8}\left[1-\left(\left(\frac{0}{3}\right)^2+\left(\frac{3}{3}\right)^2\right)\right]=0.3$$

Split value of Hair is smallest, so we will select Hair as root node.



Now we will split the remaining attributes considering Blonde data

Height->

$$\text{Split} = \frac{1}{4}\text{gini (Average)} + \frac{1}{4}\text{gini (Tall)} + \frac{2}{4}\text{gini (Short)}$$

$$= \frac{1}{4}\left[1-\left(\left(\frac{1}{1}\right)^2+\left(\frac{0}{1}\right)^2\right)\right]+\frac{1}{4}\left[1-\left(\left(\frac{0}{1}\right)^2+\left(\frac{1}{1}\right)^2\right)\right]+\frac{2}{4}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]=0.25$$

Weight->

$$\text{Split} = \frac{2}{4}\text{gini (Light)} + \frac{2}{4}\text{gini (Average)}$$

$$= \frac{2}{4}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]+\frac{2}{4}\left[1-\left(\left(\frac{1}{2}\right)^2+\left(\frac{1}{2}\right)^2\right)\right]=0.5$$

Location->

$$\text{Split} = \frac{2}{4}\text{gini (No)} + \frac{2}{4}\text{gini (Yes)}$$

$$= \frac{2}{4}\left[1-\left(\left(\frac{2}{2}\right)^2+\left(\frac{0}{2}\right)^2\right)\right]+\frac{2}{4}\left[1-\left(\left(\frac{0}{2}\right)^2+\left(\frac{2}{2}\right)^2\right)\right]=0$$

Split value of Location is smallest, so we will select Location node below Blonde branch.

From the database we can see that,

class = Yes for Hair = Blonde and Location = No , so we can directly write down 'Yes' for No branch.

class = No for Hair = Blonde and Location= Yes, so we can directly write down 'No' for Yes branch.

class = Yes for Hair = Red , so we can directly write down 'Yes' for Red branch.

class = No for Hair = Brown, so we can directly write down 'No' for Hair branch.

Final Decision Tree is,

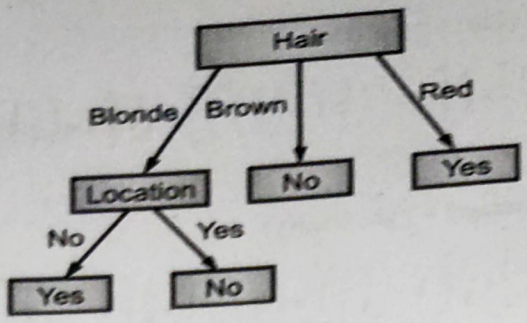


Fig. Ex. 6.5.8

UEx. 6.5.9 **MU - May 19, 10 Marks**

For a Sunburn dataset given below, construct a decision tree For the following data, Calculate Gini indexes and determines which attribute is root attribute and generate two level deep decision tree.

Sr. No.	Income	Defaulting	Credit score	Location	Give Loan?
1	Low	High	High	bad	No
2	Low	High	High	good	No
3	High	High	High	bad	Yes
4	Medium	Medium	High	bad	Yes
5	Medium	Low	Low	bad	No
6	Medium	Low	Low	good	Yes
7	High	Low	Low	good	Yes
8	Low	Medium	High	bad	No
9	Low	Low	Low	bad	No
10	Medium	Medium	Low	bad	No
11	Low	Medium	Low	good	Yes
12	High	Medium	High	good	Yes
13	High	High	Low	bad	No
14	Medium	Medium	High	good	Yes

☑ **Soln. :**

We will calculate Split for all attributes, i.e. Income, Defaulting, Creditscore and Location.

Income->

$$\text{Split} = \frac{5}{14}\text{gini (Low)} + \frac{4}{14}\text{gini (High)} + \frac{5}{14}\text{gini (Medium)}$$

$$= \frac{5}{14}\left[1-\left(\left(\frac{1}{5}\right)^2+\left(\frac{4}{5}\right)^2\right)\right]+\frac{4}{14}\left[1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right]+\frac{5}{14}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]=0.392$$

Defaulting->

$$\text{Split} = \frac{4}{14}\text{gini (High)} + \frac{6}{14}\text{gini (Medium)} + \frac{4}{14}\text{gini (Low)} = 0.438$$

Creditscore->

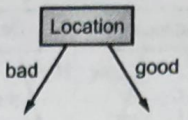
$$\text{Split} = \frac{7}{14}\text{gini (High)} + \frac{7}{14}\text{gini (Low)} = 0.493$$

Location->

$$\text{Split} = \frac{8}{14}\text{gini (bad)} + \frac{6}{14}\text{gini (good)}$$

$$= \frac{5}{8}\left[1-\left(\left(\frac{3}{5}\right)^2+\left(\frac{2}{5}\right)^2\right)\right]+\frac{3}{8}\left[1-\left(\left(\frac{0}{3}\right)^2+\left(\frac{3}{3}\right)^2\right)\right] = 0.336$$

Split value of Location is smallest, so we will select Location as root node.



Now we will split the bad branch considering remaining attributes

Income->

$$\text{Split} = \frac{3}{8}\text{gini (Low)} + \frac{2}{8}\text{gini (High)} + \frac{3}{8}\text{gini (Medium)} = 0.295$$

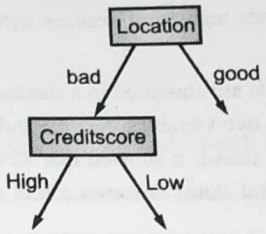
Defaulting->

$$\text{Split} = \frac{3}{8}\text{gini (High)} + \frac{3}{8}\text{gini (Medium)} + \frac{2}{8}\text{gini (Low)} = 0.34$$

Creditscore->

$$\text{Split} = \frac{4}{8}\text{gini (High)} + \frac{4}{8}\text{gini (Low)} = 0.25$$

Split value of Creditscore is smallest, so we will select Creditscore node below bad branch.



Now we will split the good branch considering remaining attributes

Income->

$$\text{Split} = \frac{2}{6}\text{gini (Low)} + \frac{2}{6}\text{gini (High)} + \frac{2}{6}\text{gini (Medium)} = 0.295$$

Defaulting->

$$\text{Split} = \frac{1}{6}\text{gini (High)} + \frac{2}{6}\text{gini (Medium)} + \frac{3}{6}\text{gini (Low)} = 0$$

Split value of Defaulting is smallest, so we will select Defaulting node below good branch

Since only one attribute is remaining, we can directly select Income below creditscore= High branch

For Location = bad and creditscore = High and Income = Low, Giveloan= No

For Location = bad and creditscore = High and Income = Medium, Giveloan= Yes

For Location = bad and creditscore = High and Income = High, Giveloan= Yes

For Location = bad and creditscore = Low, Giveloan= No

For Location = good and Defaulting = High , Giveloan= No

For Location = good and Defaulting = Low , Giveloan= Yes

For Location = good and Defaulting = Medium , Giveloan= yes

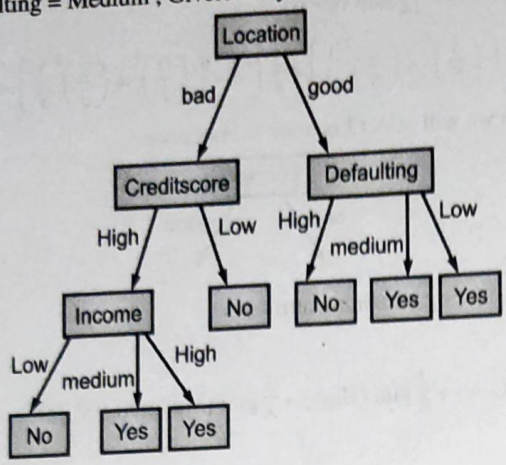


Fig. Ex. 6.5.9

6.5.5 Classification and Regression Tree (CART)

- Classification trees are used to divide the dataset into classes belonging to the target variable. Mainly the target variable has two classes that can be yes or no. When the target variable type is categorical classification trees are used.
- In certain applications the target variable is numeric or continuous in that case regression trees are used. Let's take an example of prediction of price of a flat. Hence regression trees are used for problems or tasks where we want to predict some data instead of classifying the data.
- Based on the similarity of the data the records are classified in a standard classification tree. Let's take an example of an Income tax evades. In this example we have two variables, Income and marital status that predict if a person is going to evade the income tax or not. In our training data if it showed that 85% of people who are married does not evade the income tax, we split the data here and Marital status becomes a root node in tree. Entropy or Ginny index is used in classification trees.
- The main basic working of regression tree is to fit a model. The target or response variable does not have classes so a regression model is fit using each independent variable to the target variable. Then the data is split at various split points for each independent variable. At each split point sum of squared errors (SSE) is calculated by taking the square of the difference between predicted and actual value. The criteria for root node is to select the node which is having minimum SSE among all split point errors. The further tree is built using the recursive procedure.

6.5.6 Example of Regression Tree

Ex. 6.5.10

Buying_Price	Lug_Boot	Safety	Maintenance_ Price? (in thousand)
Low	Small	High	25
Low	Small	Low	30
Medium	Small	High	46
High	Small	High	45
High	Big	High	52
High	Big	Low	23
Medium	Big	Low	43

Buying_Price	Lug_Boot	Safety	Maintenance_ Price? (in thousand)
Low	Small	High	35
Low	Big	High	38
High	Big	High	46
Low	Big	Low	48
Medium	Small	Low	52
Medium	Big	High	44
High	Small	Low	30

n. :

dard deviation

A decision tree is built up top down from root node and involved partitioning the data into subsets that contain instances with similar values. We use SD to calculate homogeneity of a numerical sample.

$$SD, S = \sqrt{\frac{\sum (x-\mu)^2}{n}} = 9.32$$

SD Reduction

It is based on the decrease in SD after a dataset is split on an attribute. Constructing a tree is all about finding attribute that returns highest SDR.

- **Step 1 :**

SD (Maintenance_Price?) = 9.32

- **Step 2 :**

The dataset is then split on the different attribute.SD for each branch is calculated. The resulting SD is subtracted from SD before split.

		Maintenance_ Price(SD)
Buying_Price	Low	7.78
	Medium	3.49
	High	10.87
SD (Maintenance_Price, Buying_Price) = P (Low) SD (Low) + P (Medium) SD (Medium) + P (High) SD (High) $= \frac{5}{14} \times 7.78 + \frac{4}{14} \times 3.49 + \frac{5}{14} \times 10.87 = 7.66$ SDR = SD (Maintenance_ Price) – SD (Maintenance_ Price, Buying_Price) = 9.32 – 7.66 = 1.66		

		Maintenance_ Price (SD)
Lug_Boot	Small	9.36
	Big	8.37
SD (Maintenance_ Price, Lug_Boot) = P (Small) SD (Small) + P (Big) SD (Big) $= \frac{7}{14} \times 9.36 + \frac{7}{14} \times 8.37 = 8.86$ SDR = SD (Maintenance_ Price) – SD (Maintenance_ Price, Lug_Boot) = 9.32 – 8.86 = 0.46		

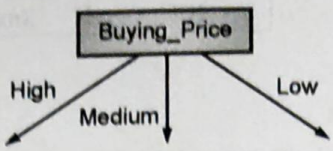
		Maintenance_ Price (SD)
Safety	High	7.87
	Low	10.59

SD (Maintenance_ Price, Safety) = P (High) SD (High) + P (Low) SD (Low)

$= \frac{8}{14} \times 7.87 + \frac{6}{14} \times 10.59 = 9.02$

SDR = SD (Maintenance_ Price) – SD (Maintenance_ Price, Safety) = 9.32 – 9.02 = 0.3

SDR of Buying_Price is highest so we select Buying_Price as our root node.



To avoid over fitting we should terminate unnecessary building branches. For example if there are less than five instances in the sub data set or standard deviation can be less than 5% of the entire data set. I prefer to apply the first one. I will terminate the branch if there are less than 5 instances in current sub data set. If this termination condition is satisfied then i will calculate average of sub data set.

- **Step 2(a) :**

Now we will consider the records of 'High'.

Buying_Price	Lug_Boot	Safety	Maintenance_ Price? (in thousand)
High	Small	High	45
High	Big	High	52
High	Big	Low	23
High	Big	High	46
High	Small	Low	30

For Buying_Price = High, SD = 10.87

We will calculate SDR of only Lug_Boot and Safety

		Maintenance_ Price (SD)
Lug_Boot	Small	7.5
	Big	12.49

SD(High, Lug_Boot) = P (Small) SD (Small) + P (Big) SD (Big)

$= \frac{2}{5} \times 7.5 + \frac{3}{5} \times 12.49 = 10.49$

SDR = SD(High) – SD (Maintenance_ Price, Lug_Boot) = 10.87 – 10.49 = 0.38

		Maintenance_ Price (SD)
Safety	High	3.09
	Low	3.50

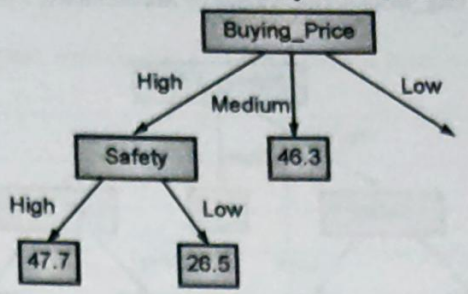
SD (High, Safety) = P (High) SD (High) + P (Low) SD (Low)

$= \frac{3}{5} \times 3.09 + \frac{2}{6} \times 3.5 = 3.25$

SDR = SD (High) – SD (Maintenance_ Price, Safety) = 10.87 – 3.25 = 7.62

SDR of Safety is highest so we select Safety as next node below High branch.

- For Buying_Price = High and Safety = High, we can directly write down the answer.
- For Buying_Price = High and Safety = Low, we can directly write down the answer.



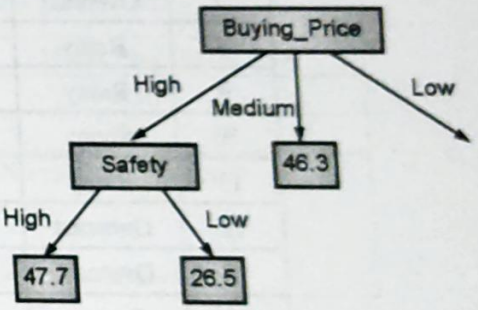
- To write down the answer we take average of values of following records,
  - For Buying_Price = High and Safety = High, Maintenance _Price = (45 + 52 + 46) /3 = 47.7
  - For Buying_Price = High and Safety = Low, Maintenance _Price = (23 + 30) /2 = 26.5

Step 3:

Now we will consider the records of 'Medium'

Buying_Price	Lug_Boot	Safety	Maintenance_ Price? (in thousand)
Medium	Small	High	46
Medium	Big	Low	43
Medium	Small	Low	52
Medium	Big	High	44

For Buying_Price = Medium, we can directly write down the answer as 46.3. The answer is calculated by taking the average of values of Maintenance_ Price for Medium records (average of 46, 43, 52, and 44).



- **Step 4 :**

Now we will consider the records of 'Low'.

Buying_Price	Lug_Boot	Safety	Maintenance_ Price? (in thousand)
Low	Small	High	25
Low	Small	Low	30
Low	Small	High	35
Low	Big	High	38
Low	Big	Low	48

For Buying_Price = Low, SD = 7.78

- Now only Lug_Boot attribute is remaining, so we can directly take Lug_Boot as a next node below Low branch.

- To write down the answer we take average of values of following records,
  - For Buying_Price = Low and Lug_Boot = Small, Maintenance_Price = (25 + 30 + 35) /3 = 30
  - For Buying_Price = Low and Lug_Boot = Big, values of Maintenance_Price = (38 + 48) /2 = 43.
- Final Regression Tree is,

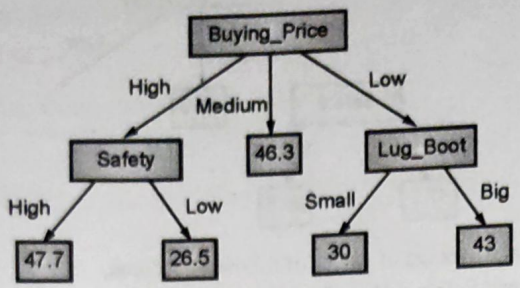


Fig. Ex. 6.5.10

Ex. 6.5.11

Day	Outlook	Temp	Humidity	Windy	Hours Play?
1	Rainy	Hot	High	False	25
2	Rainy	Hot	High	True	30
3	Overcast	Hot	High	False	46
4	Sunny	Mild	High	False	45
5	Sunny	Cool	Normal	False	52
6	Sunny	Cool	Normal	True	23
7	Overcast	Cool	Normal	True	43
8	Rainy	Mild	High	False	35
9	Rainy	Cool	Normal	False	38
10	Sunny	Mild	Normal	False	46
11	Rainy	Mild	Normal	True	48
12	Overcast	Mild	High	True	52
13	Overcast	Hot	Normal	False	44
14	Sunny	Mild	High	True	30

☑ **Soln. :**

☞ **Standard deviation**

A decision tree is built up top down from root node and involved partitioning the data into subsets that contain instances with similar values. We use SD to calculate homogeneity of a numerical sample.

$$SD, S = \sqrt{\frac{\sum (x-\mu)^2}{n}} = 9.32$$

☞ **SD Reduction**

It is based on the decrease in SD after a dataset is split on an attribute. Constructing a tree is all about finding attribute that returns highest SDR.

- **Step 1 :**

SD (Hours Play?) = 9.32

- **Step 2 :**

The dataset is then split on the different attribute.SD for each branch is calculated. The resulting SD is subtracted from SD before split.

		Hours Play(SD)
Outlook	Rainy	7.78
	Overcast	3.49
	Sunny	10.87

SD (HoursPlay, Outlook) = P (Rainy) SD (Rainy) + P (Overcast) SD (Overcast) + P (Sunny) SD (Sunny)

$= \frac{5}{14} \times 7.78 + \frac{4}{14} \times 3.49 + \frac{5}{14} \times 10.87 = 7.66$

SDR = SD (Hours Play) – SD (Hours Play, Outlook) = 9.32 – 7.66 = 1.66

		Hours Play(SD)
Temp	Cool	10.51
	Hot	8.95
	Mild	7.65

SD(Hours Play, Temp) = P (Cool) SD (Cool) + P (Hot) SD (Hot) + P (Mild) SD (Mild)

$= \frac{4}{14} \times 10.51 + \frac{4}{14} \times 8.95 + \frac{6}{14} \times 7.65 = 8.84$

SDR = SD (Hours Play) – SD(Hours Play, Temp) = 9.32 – 8.84 = 0.48

		Hours Play (SD)
Humidity	High	9.36
	Normal	8.37

SD (Hours Play, Humidity) = P (High) SD (High) + P (Normal) SD (Normal)

$= \frac{7}{14} \times 9.36 + \frac{7}{14} \times 8.37 = 8.86$

SDR = SD (Hours Play) – SD (Hours Play, Humidity) = 9.32 – 8.86 = 0.46

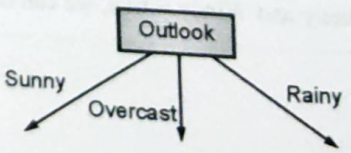
		Hours Play(SD)
Windy	False	7.87
	True	10.59

SD (HoursPlay, Windy) = P (False) SD (False) + P (True) SD (True)

$= \frac{8}{14} \times 7.87 + \frac{6}{14} \times 10.59 = 9.02$

SDR = SD (Hours Play) – SD (Hours Play, Windy) = 9.32 – 9.02 = 0.3

SDR of outlook is highest so we select outlook as our root node.



Step 2(a) :

Now we will consider the records of 'sunny'.

Day	Outlook	Temp	Humidity	Windy	Hours Play?
4	Sunny	Mild	High	False	45
5	Sunny	Cool	Normal	False	52
6	Sunny	Cool	Normal	True	23
10	Sunny	Mild	Normal	False	46
14	Sunny	Mild	High	True	30

For Outlook = Sunny, SD = 10.87

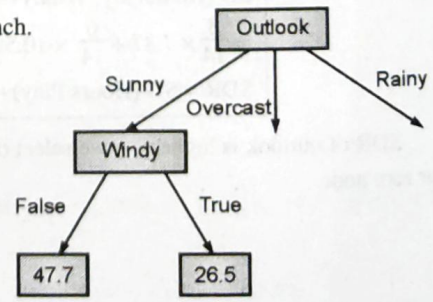
We will calculate SDR of only Temp, Humidity and Windy

		Hours Play (SD)
Temp	Cool	14.5
	Mild	7.31
SD (Sunny, Temp) = P (Cool) SD (Cool) + P (Mild) SD (Mild) $=\frac{2}{5}\times 14.5+\frac{3}{5}\times 7.31 = 10.18$ SDR = SD(Sunny) – SD (Hours Play, Temp) = 10.87 – 10.18 = 0.69		

		Hours Play (SD)
Humidity	High	7.5
	Normal	12.49
SD (Sunny, Humidity) = P (High) SD (High) + P (Normal) SD (Normal) $=\frac{2}{5}\times 7.5+\frac{3}{5}\times 12.49 = 10.49$ SDR = SD (Sunny) – SD (Hours Play, Humidity) = 10.87 – 10.49 = 0.38		

		Hours Play (SD)
Windy	False	3.09
	True	3.50
SD (Sunny, Windy) = P (False) SD (False) + P (True) SD (True) $=\frac{3}{5}\times 3.09+\frac{2}{5}\times 3.5 = 3.25$ SDR = SD (Sunny) – SD (Hours Play, Windy) = 10.87 – 3.25 = 7.62		

- SDR of windy is highest so we select windy as next node below sunny branch.
  - For Outlook = sunny and Windy = false, we can directly write down the answer.
  - For Outlook = sunny and Windy = true, we can directly write down the answer



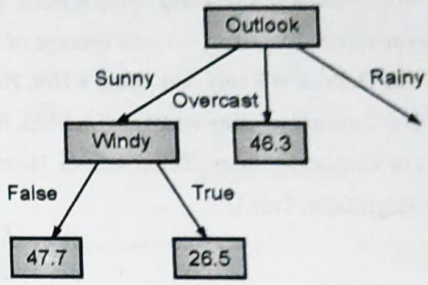
- To write down the answer we take average of values of following records,
  - For Outlook = sunny and Windy = false, Hours Play = (45 + 52 + 46) /3 = 47.7
  - For Outlook = sunny and Windy = true, Hours Play = (23 + 30) /2 = 26.5

Step 3 :

Now we will consider the records of 'overcast'. For Overcast we will directly write down the answer.

Day	Outlook	Temp	Humidity	Windy	Hours Play?
3	Overcast	Hot	High	False	46
7	Overcast	Cool	Normal	True	43
12	Overcast	Mild	High	True	52
13	Overcast	Hot	Normal	False	44

The answer is calculated by taking the average of values of Hours Play for overcast records (average of 46, 43, 52, and 44).



Step 4:

Now we will consider the records of 'Rainy'.

Day	Outlook	Temp	Humidity	Windy	Hours Play?
1	Rainy	Hot	High	False	25
2	Rainy	Hot	High	True	30
8	Rainy	Mild	High	False	35
9	Rainy	Cool	Normal	False	38
11	Rainy	Mild	Normal	True	48

For Outlook = Rainy, SD = 7.78

Now we will calculate SDR of only Humidity and Temp

		Hours Play (SD)
Temp	Hot	2.5
	Cool	0
	Mild	6.5

SD (Rainy, Temp) = P (Hot) SD (Hot) + P (Cool) SD (Cool) + P (Mild) SD (Mild)

$= \frac{2}{5} \times 2.5 + \frac{1}{5} \times 0 + \frac{2}{5} \times 6.5 = 3.6$

SDR = SD (Rainy) – SD (Rainy Play, Temp) = 7.78 – 3.6 = 4.18

		Hours Play(SD)
Humidity	High	5
	Normal	5
SD (Sunny, Humidity) = P (High) SD (High) + P (Normal) SD (Normal) $= \frac{3}{5} \times 5 + \frac{2}{5} \times 5 = 5$		
SDR = SD (Rainy) – SD (Hours Play, Humidity) = 7.78 – 5 = 2.28		

- SDR of Temp is highest so we select Temp as next node below rainy branch.
  - For Outlook = Rainy and Temp = Cool, we can directly write down the answer.
  - For Outlook = Rainy and Temp = Hot, we can directly write down the answer.
  - For Outlook = Rainy and Temp = Mild, we can directly write down the answer.
- To write down the answer we take average of values of following records,
  - For Outlook = Rainy and Temp = Hot, Hours Play = (25 + 30) /2 = 27.5
  - For Outlook = Rainy and Temp = Mild, Hours Play = (35 + 48) /2 = 41.5
  - For Outlook = Rainy, Temp = Cool, Hours Play = 38
- Final Regression Tree is,

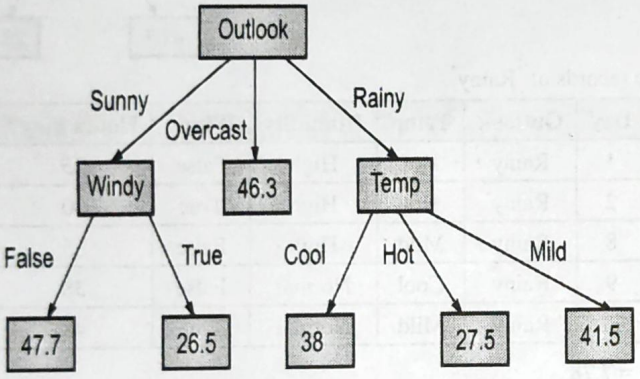


Fig. Ex. 6.5.11

6.6 SUPERVISED LEARNING : SUPPORT VECTOR MACHINE

6.6.1 Maximum Margin Linear Separators

UQ. What is SVM ? Explain the following terms: separating hyperplane, margin and support vectors with suitable example. **(MU - May 15, 4 Marks)**

UQ. What are the key terminologies of Support Vector Machine ? **(MU - May 16, 5 Marks)**

UQ. What is Support Vector Machine ? **(MU - May 17, Dec. 19, 4 Marks)**

UQ. Illustrate Support Vector machine with neat labeled sketch. **(MU - May 19, 4 Marks)**

- Support Vector Machine is a type of supervised learning that can be used for classification or regression. Even if the data points are unseen (not from the training dataset), support vector machine classifies the data properly.
- Let's take an example of dataset that belongs to two different categories, and the distribution of data is proper means the data is separated from each other properly.

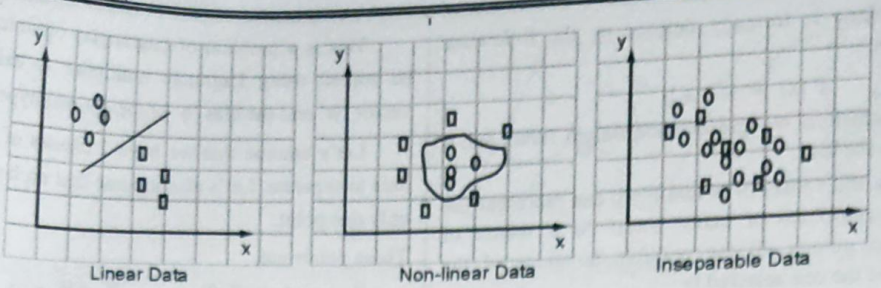


Fig. 6.6.1 : Different types of data

- In this case we can draw a straight line (decision dary) on the graph in such a way that the input e is divided in to two regions.
- ata points that belongs to one category lies on one side of the decision boundary and the data points of other category lies on the opposite side.
- Such type of data is called as linearly separable data.
- **Separating hyperplane** is the line which is used to separate the dataset. If we are using simple 2- dimensional plots then it's just a line. We require a plane to separate the data if data is 3 dimensional. So we can say that if data is N dimensional, we require N-1 dimensional hyperplane.
- We want that our classifier should be designed in a manner that if a data point is far away from the decision boundary then we will be more confident about the prediction we have made.
- We would like to find the data point near to the separating hyperplane and also make sure that this point should be far away from the separating line as possible.
- This is called as **margin**. We would like to find the greatest possible margin, because if we trained our classifier on limited data or made a mistake, we would want it to be as robust as possible.
- **Support vectors** are the points which are nearest to the separating hyperplane. We have to maximize the distance between the support vectors and the separating line.

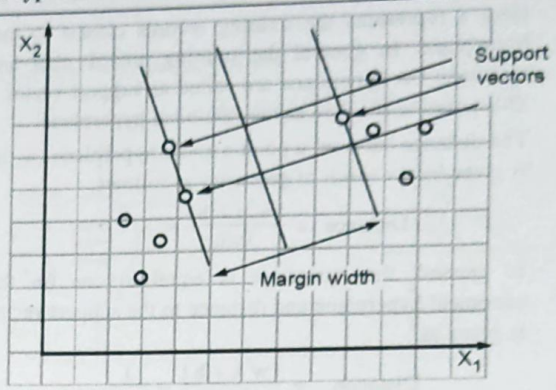


Fig. 6.6.2 : Support vectors and Margin

Distance between a hyperplane (w, b) and a point x is calculated as,

$$\text{Distance} = \frac{|w^t x + b|}{\|w\|}$$

6.6.2 Quadratic Programming Solution to Find Maximum Margin Separator

UQ. Explain the term: hyperplane with suitable example. **(MU - May 15, 1 Marks)**

UQ. Write detail notes on: Quadratic Programming solution for finding maximum margin separation in support vector machine. **(MU - May 16, 10 Mark)**

UQ. How to compute the margin ? **(MU - May 17, 6Marks)**

UQ. Explain how margin is computed and optimal hyper-plane is decided ? **(MU - Dec. 19, 6 Marks)**

UQ. Show how to derive optimal hyper-Plane? **(MU - May 19, 6 Marks)**

- A **hyperplane** is formally defined by the following notation as,

$$F(x) = w^t x + b$$

In above equation, w represents the weight vector and b represents the bias.

- By scaling the values of w and b we can represent the optimal hyperplane in many ways. As a matter of convention among all the possible notations of the hyperplane the one selected is

$$|w^t x + b| = 1$$

- Here x represents the training records closest to the hyperplane. In general the training records that are closest to the hyperplane are called as support vectors. This notation is called as the canonical hyperplane.
- The distance between a point x and a hyperplane (w, b) is given by the result of geometry as follows,

$$\text{Distance} = \frac{|w^t x + b|}{\|w\|}$$

- In general, the numerator is equal to one for the canonical hyperplane and distance to the support vector is given as,

$$\text{Distance}_{sv} = \frac{|w^t x + b|}{\|w\|} = \frac{1}{\|w\|}$$

- Margin is twice the distance to nearest samples

$$M = \frac{2}{\|w\|}$$

- Ultimately, the task of maximizing M is same as compared to the task of minimizing a function L (w) subject to some conditions. The conditions used to model the requirements for correct classification of all training samples x_i by the hyperplane are formally stated as,

$$\min_{w, b} L(w) = \frac{1}{2}\|w\|^2 \text{ subject to } y_i (w^t x + b) \geq 1 \text{ for all i.}$$

Where y_i represents the labels of training

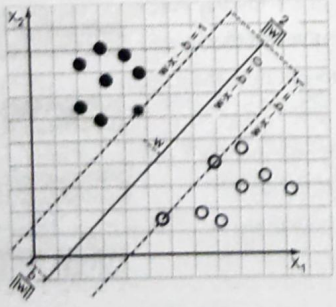


Fig. 6.6.3 : Solution to find maximum margin

This is a problem of Lagrangian optimization that can be solved using Lagrange multiplier to calculate weight vector 'w' and the bias 'b' of the optimal hyperplane.

Let's assume that we have 2 classes of 2 dimensional data to separate. Let's also assume that each class consist of only one point.

These points are,

$$X_1 = A_1 = (3, 3) \; ; \qquad X_2 = B_1 = (6, 6)$$

Find the hyper plane that separates theses 2 classes

$$f(w) = \frac{1}{2}\|w\|^2$$

The constraints are,

$$c_1 (w, b) = y_i |wx_i + b| - 1 \geq 0$$

$$c_1 (w, b) = 1 |wx_1 + b| - 1 \geq 0$$

$$c_2 (w, b) = -1 |wx_2 + b| - 1 \geq 0$$

Next, we put equation into form of Lagrangian

$$L (w, b, m) = f(w) - m_1 c_1 (w, b) - m_2 c_2 (w, b)$$

$$= \frac{1}{2}\|w\|^2 - m_1 ((wx_1 + b) - 1) - m_2 (-(wx_2 + b) - 1)$$

$$= \frac{1}{2}\|w\|^2 - m_1 ((wx_1 + b) - 1) + m_2 ((wx_2 + b) + 1)$$

We solve for the gradient of Lagrangian

$$\nabla L (w, b, m) = \nabla f(w) - m_1 \nabla c_1 (w, b) + m_2 \nabla c_2 (w, b) = 0$$

$$\frac{\partial}{\partial w} L (w, b, m) = w - m_1 x_1 + m_2 x_2 = 0 \quad ...(6.6.1)$$

$$\frac{\partial}{\partial b} L (w, b, m) = -m_1 + m_2 = 0 \quad ...(6.6.2)$$

$$\frac{\partial}{\partial \lambda_1} L (w, b, m) = (wx_1 + b) - 1 = 0 \quad ...(6.6.3)$$

$$\frac{\partial}{\partial \lambda_2} L (w, b, m) = (wx_2 + b) + 1 = 0 \quad ...(6.6.4)$$

Equating Equation (6.6.3) and (6.6.4), we get

$$(wx_1 + b) - 1 = (wx_2 + b) + 1$$

$$(wx_1) - 1 = (wx_2) + 1$$

$$(wx_1) - (wx_2) = 2$$

$$w (x_1 - x_2) = 2$$

w is divided into parts as,

$$w = (w_1, w_2)$$

$$W(x_1 - x_2) = 2$$

$$(w_1, w_2)[(3, 3) - (6, 6)] = 2$$

$$(w_1, w_2)[(-3, -3)] = 2$$

$$-3w_1 - 3w_2 = 2$$

$$w_1 = -(0.67 + w_2) \quad ...(6.6.5)$$

Adding values to Equation (6.6.1) and combining with Equation (6.6.2)

$$(w_1, w_2) - m_1(1, 1) + m_2(2, 2) = 0$$

From Equation (6.6.2)

$$m_1 = m_2$$

$$(w_1, w_2) - m_1(3, 3) + m\lambda_1(6, 6) = 0$$

$$(w_1, w_2) + m_1(3, 3) = 0$$

$$w_1 + 3m_1 = 0 \quad ...(6.6.6)$$

$$w_2 + 3m_1 = 0 \quad ...(6.6.7)$$

Equating these we get, $w_1 = w_2$

Putting this in Equation (6.6.5)

$$w_1 = w_2 = -0.34$$

Putting this in either Equation (6.6.6) or Equation (6.6.7) will give

$$m_1 = m_2 = 0.11$$

And finally, using this in Equation (6.6.3) and ation (6.6.4)

$$b = 1 - (wx_1) \text{ or } = -1 - (w x_2)$$

$$= 1 - ((-0.34, -0.34), (3, 3)) \text{ or}$$

$$= 1 - ((-0.34, -0.34), (6, 6)) = 3.04$$

6.6.3 Kernels for Learning Non-Linear Functions

- Linear classifiers are able to separate only linearly separable data. Support vector machine provides the solution to this problem by transforming an input space into a feature space that contains non linear features.
- A hyperplane is constructed in the feature space so that other equations remain the same. This is also known as non-linear support vector machine. Here we separate the data linearly using a high dimensional space.
- Kernel functions with its own set of variables are used for this purpose. The result is going to be non linear if we convert this back to the original feature space.

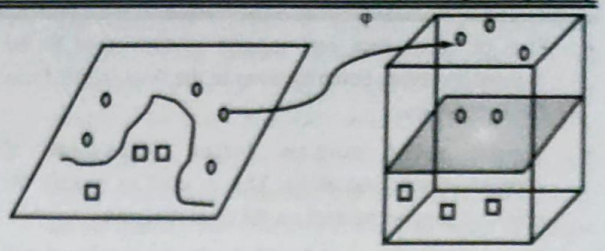


Fig. 6.6.4 : Mapping of Input space to Feature space

Particularly, the data is preprocessed with

$$X \rightarrow \Phi(x)$$

And then $\Phi(x)$ is mapped to y

$$F(x) = w\,\Phi(x) + b$$

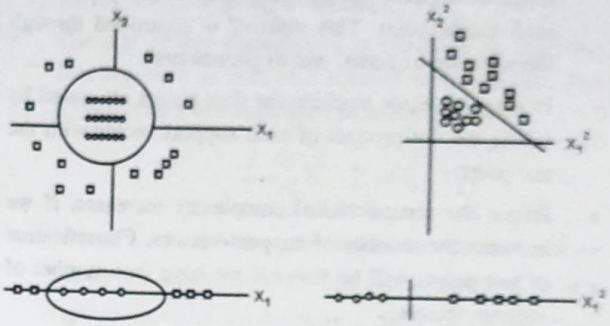


Fig. 6.6.5 : Mapping of one feature space to another feature space

- Kernel function transforms the data in to an easily understandable form. This is done via mapping input space to another feature space.
- In support vector machine inner product is calculated of two vectors and the result of this is always a single number. When we replace this inner product by a kernel it is called as kernel trick.
- There are different algorithms that use different kinds of kernel functions.
- Among the different types of kernel functions such as nonlinear, linear, radial basis function (RBF), polynomial, and sigmoid, RBF is mostly used. The reason for this is along the X-axis radial basis function gives the localized and finite response.
- The number of support vectors will be determined based on the different criteria's such as what is the complexity of the model, how much slack is allowed.

Module 6

- One or more than one support vectors need to be defined for every complications in the final model from the input space.
- Support vector machines output compromises of support vectors and alpha. This is used to specify the effect of support vectors on the final decision.
- If we select the model with high complexity it will result in to over fitting. For better generalization if large margin is selected then it may lead to incorrect classification.
- And accuracy depends on the trade-off between these two selections criteria. If we over fit the data then the range of support vectors may vary from very less to each single point. This tradeoff is controlled through the selection of kernel and its parameters.
- In support vector machine the data points are tested by taking the dot product of each support vector with the test point.
- Hence the computational complexity increases, if we increase the number of support vectors. Classification of test points will be faster if we have less number of support vectors.

6.6.4 Rules for the Kernel Function

Kernel function or a window is defined as follows:

If $\| \bar{x} \| \leq 1$ then $K(\bar{x}) = 1$ else 0.

This kernel function is shown by the Fig. 6.6.6,

$$K((z - x_i)/h) = 1$$

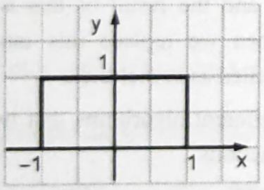


Fig. 6.6.6

For a fixed value of x_i, the function takes the value as 1 as shown in Fig. 6.6.7.

By selecting the argument of K (·), window can be moved to be centerd at the point x_i and to be of radius h.

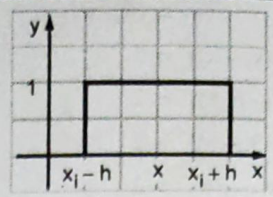


Fig. 6.6.7

6.6.5 Different Types of SVM Kernels

1. Polynomial kernel

Polynomial kernel is mostly used in image processing methods.

Polynomial Kernel is represented as,

$$K(x_j, x_k) = (x_j \cdot x_k + 1)^p$$

Here p represents the degree of the polynomial.

2. Gaussian kernel

There are some applications where prior knowledge is not available. For this type of applications Gaussian kernel is used.

Gaussian kernel is defined as,

$$K(x, y) = \exp\left(-\frac{\| x - y \|^2}{2\sigma^2}\right)$$

3. Gaussian radial basis function (RBF)

This is also used for the applications where prior knowledge is not available.

Gaussian radial basis function is defined as,

$$K(x_i, x_j) = \exp(-\gamma \| x - y \|^2) \text{ for } \gamma > 0$$

Sometimes it is parametrized using the value of γ as $1/2\sigma^2$

4. Laplace RBF kernel

Laplace RBF kernel is defined as,

$$K(x, y) = \exp\left(-\frac{\| x - y \|}{\sigma}\right)$$

5. Hyperbolic tangent kernel

Hyperbolic tangent kernel is used in neural networks.

It is defined as,

$K(x_i, x_j) = \tanh(kx_i * x_j + c)$, for some (not every) $k > 0$ and $c < 0$.

6. Sigmoid kernel

Sigmoid kernel can be used as a proxy for neural networks. It is defined as,

$$K(x, y) = \tanh(\alpha x^d y + c)$$

7. Bessel function of the first kind Kernel

Cross terms in mathematical functions can be removed by using this type of kernel function.

It is defined as,

$$K(x, y) = \frac{J_{v+1}(\sigma \| x - y \|)}{\| x - y \|^{-n(v+1)}}$$

Here j represents the Bessel function of first type.

ANOVA radial basis kernel

In regression problems this kernel can be used.

It is defined as,

$$K(x, y) = \sum_{k=1}^{n} \exp(-\sigma (x^k - y^k)^2)^d$$

6.7 UNSUPERVISED LEARNING : K MEANS CLUSTERING

- In unsupervised learning the most important task is the Clustering. Clustering is used to store data points in to related groups. In clustering advance knowledge is not present about the group definitions.

Definition : "Clustering is a process of partitioning a set of data in a set of meaningful sub-classes, called as clusters".

- In clustering we group the "similar" objects in one cluster and "dissimilar" objects in another cluster.

K- means Clustering

- To solve the well known clustering problem K-means is used, which is one of the simplest unsupervised learning algorithms.
- Given data set is classified assuming some prior number of clusters through a simple and easy procedure. In k- means clustering for each cluster one centroid is defined. Total there are k centroids.
- The centroids should be defined in a tricky way because result differs based on the location of centroids. To get the better results we need to place the centroids far away from each other as much as possible.
- Next, each point from the given data set is stored in a group with closest centroid. This process is repeated for all the points. The first step is finished when all points are grouped. In the next step new k centroids are calculated again from the result of the earlier step.
- After finding these new k centroids, a new grouping is done for the data points and closest new centroids. This process is done iteratively.
- The process is repeated unless and until no data point moves from one group to another.
- The aim of this algorithm is to minimize an objective function such as sum of a squared error function. The objective function is defined as follows :

$$J = \sum_{j=1}^{k} \sum_{i=1}^{x} \| x_i^j - C_j \|^2$$

- Here $\| x_i^j - C_j \|^2$ shows the selected distance measure between a data point x_i^j and the cluster centre C_j. It is a representation of the distance of the n data points from their respective cluster centers.

UQ. Describe the essential steps of K-means algorithm for clustering analysis. **(MU - May 15, 5 Marks)**

- The algorithm is comprises of the following steps :
  1. Identify the K centroids for the given data points that we want to cluster.
  2. Store each data point in the group that has the nearest centroid.
  3. When all data points have been stored, redefine the K centroids.
  4. Repeat Steps 2 and 3 until the no data points move from one group to another. The result of this process is the clusters from which the metric to be minimized can be calculated.
- The k-means algorithm does not guarantee the most optimal solution corresponding to global minimum objective function, although it can be proved that the process will always terminate.

Module 6

- Initial random selection of cluster centers affects the performance of the algorithm. The k-means algorithm is applied for a number of times to reduce this effect.
- Let's assume that n sample data points $x_1, x_2, ..., x_n$ of the same class are present, and we know that the data points belongs to k clusters, $k < n$.
- Let m_i represents the mean of the data points in cluster i. x can be stored in cluster i, if $\| x - m_i \|$ is the minimum of all the k distances.
- The k-means procedure is shown below:
- Select initial values for the means $m_1, m_2, ..., m_k$

  Until no data point moves from one group to another
  - Use the calculated means to group the data points into clusters
  - For i from 1 to k

    Mean of all of the samples for cluster i is used to replace m_i with the
  - end_for

  end_until
- The K-means algorithm is implemented in three steps.
- Iterate until stable (= no data point move group)
  1. Determine the centroid coordinate
  2. Determine the distance of each data point to the centroid
  3. Group the data points based on minimum distance.

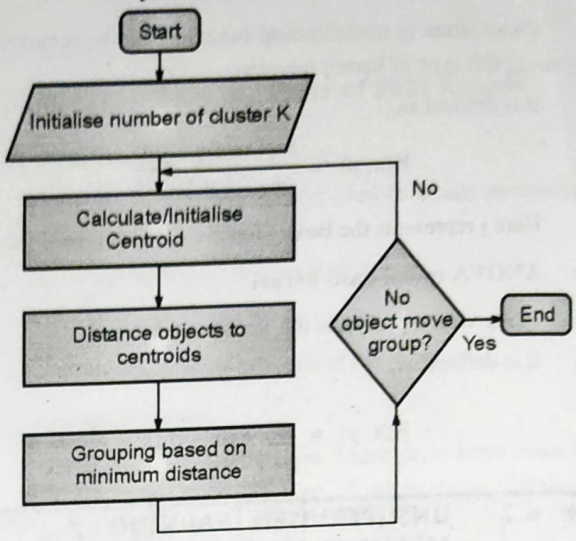


Fig. 6.7.1

6.7.1 Examples on K-means Clustering

Ex. 6.7.1 : Given { 2, 4, 10, 12, 3, 20, 30, 11, 25}. Assume number of clusters i.e. K = 2

Soln. :

Randomly assign means : $m_1 = 3$, $m_2 = 4$

The numbers which are close to mean $m_1 = 3$ are grouped into cluster k_1 and others in k_2.

Again calculate new mean for new cluster group.

$K_1 = (2, 3\}, k_2 = \{4, 10, 12, 20, 30, 11, 25\}$ $m_1 = 2.5, m_2 = 16$

$K_1 = (2, 3, 4\}, k_2 = \{10, 12, 20, 30, 11, 25\}$ $m_1 = 3, m_2 = 18$

$K_1 = (2, 3, 4, 10\}, k_2 = \{12, 20, 30, 11, 25\}$ $m_1 = 4.75, m_2 = 19.6$

$K_1 = (2, 3, 4, 10, 11, 12\}, \quad k_2 = \{20, 30, 25\}$ $m_1 = 7, m_2 = 25$

Final clusters

$K_1 = (2, 3, 4, 10, 11, 12\}, \quad k_2 = \{20, 30, 25\}$

Ex. 6.7.2 : Given {10, 4, 2, 12, 3, 20, 30, 11, 25, 31} Assume number of clusters i.e. K = 2

Soln. :

Randomly assign alternative values to each cluster

$K_1 = (10, 2, 3, 30, 25\}, \quad k_2 = \{4, 12, 20, 11, 31\}$ $m_1 = 14, \quad m_2 = 15.6$

Re assign

K_1 = (2, 3, 4, 10, 11, 12}, k_2 = {20, 25, 30, 31} m_1 = 7, m_2 = 26.5

Re assign

K_1 = (2, 3, 4, 10, 11, 12}, k_2 = {20, 25, 30, 31} m_1 = 7, m_2 = 26.5

Final clusters

K_1 = (2, 3, 4, 10, 11, 12}, k_2 = {20, 25, 30, 31}

Ex. 6.7.3 : Let's assume that we have 4 types of items and each item has 2 attributes or features. We need to group these items in to k = 2 groups of items based on the two features.

Object	Attribute 1(x) Number of parts	Attribute 2(y) Colour code
Item 1	1	1
Item 2	2	1
Item 3	4	3
Item 4	5	4

☑ **Soln. :**

Initial value of centroid

Suppose we use item 1 and 2 as the first centroids, $c_1 = (1, 1)$ and $c_2 = (2, 1)$

The distance of item 1 = (1, 1) to $c_1 = (1, 1)$ and with $c_2 = (2, 1)$ is calculated as,

$$D = \sqrt{(1-1)^2 + (1-1)^2} = 0$$

$$D = \sqrt{(1-2)^2 + (1-1)^2} = 1$$

The distance of item 2 = (2, 1) to $c_1 = (1, 1)$ and with $c_2 = (2, 1)$ is calculated as,

$$D = \sqrt{(2-1)^2 + (1-1)^2} = 1$$

$$D = \sqrt{(2-2)^2 + (1-1)^2} = 0$$

The distance of item 3 = (4, 3) to $c_1 = (1, 1)$ and with $c_2 = (2, 1)$ is calculated as,

$$D = \sqrt{(4-1)^2 + (3-1)^2} = 3.61$$

$$D = \sqrt{(4-2)^2 + (3-1)^2} = 2.83$$

The distance of item 4 = (5, 4) to $c_1 = (1, 1)$ and with $c_2 = (2, 1)$ is calculated as,

$$D = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$D = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Objects-centroids distance

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{matrix} c_1 = (1, 1) & \text{group 1} \\ c_2 = (2, 1) & \text{group 2} \end{matrix}$$

To find the cluster of each item we consider the minimum Euclidian distance between group1 and group 2.

From the above object centroid distance matrix we can see,

- Item 1 has minimum distance for group1, so we cluster item 1 in group 1.
- Item 2 has minimum distance for group 2, so we cluster item 2 in group 2.
- Item 3 has minimum distance for group 2, so we cluster item 3 in group 2.
- Item 4 has minimum distance for group 2, so we cluster item 4 in group 2.

Object Clustering

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Iteration 1 : Determine centroids

C_1 has only one member thus $c_1 = (1, 1)$ remains same.

$$C_2 = (2 + 4 + 5/3, 1 + 3 + 4/3) = (11/3, 8/3)$$

The distance of item 1 = (1, 1) to $c_1 = (1, 1)$ and with $c_2 = (11/3, 8/3)$ is calculated as,

$$D = \sqrt{(1-1)^2 + (1-1)^2} = 0$$

$$D = \sqrt{(1-11/3)^2 + (1-8/3)^2} = 3.41$$

The distance of item 2 = (2, 1) to $c_1 = (1, 1)$ and with $c_2 = (11/3, 8/3)$ is calculated as,

$$D = \sqrt{(2-1)^2 + (1-1)^2} = 1$$

$$D = \sqrt{(2-11/3)^2 + (1-8/3)^2} = 2.36$$

The distance of item 3 = (4, 3) to $c_1 = (1, 1)$ and with $c_2 = (2, 1)$ is calculated as,

$$D = \sqrt{(4-1)^2 + (3-1)^2} = 3.61$$

$$D = \sqrt{(4-11/3)^2 + (3-8/3)^2} = 0.47$$

The distance of item 4 = (5, 4) to $c_1 = (1, 1)$ and with $c_2 = (2, 1)$ is calculated as,

$$D = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$D = \sqrt{(5-11/3)^2 + (4-8/3)^2} = 1.89$$

Objects-centroids distance

$$D^2 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.41 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} c_1 = (1, 1) & \text{group 1} \\ c_2 = \left(\frac{11}{3}, \frac{8}{3}\right) & \text{group 2} \end{matrix}$$

From the above object centroid distance matrix we can see,

- Item 1 has minimum distance for group1, so we cluster item 1 in group 1.
- Item 2 has minimum distance for group 1, so we cluster item 2 in group 1.
- Item 3 has minimum distance for group 2, so we cluster item 3 in group 2.
- Item 4 has minimum distance for group 2, so we cluster item 4 in group 2.

Object Clustering

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Iteration 2 : Determine centroids

$$C_1 = (1 + 2/2, 1 + 1/2) = (3/2, 1)$$

$$C_2 = (4 + 5/2, 3 + 4/2) = (9/2, 7/2)$$

The distance of item 1 = (1, 1) to $c_1 = (3/2, 1)$ and with $c_2 = (9/2, 7/2)$ is calculated as,

$$D = \sqrt{(1-3/2)^2 + (1-1)^2} = 0.5$$

$$D = \sqrt{(1-9/2)^2 + (1-7/2)^2} = 4.3$$

The distance of item 2 = (2, 1) to $c_1 = (3/2, 1)$ and with $c_2 = (9/2, 7/2)$ is calculated as,

$$D = \sqrt{(2-3/2)^2 + (1-1)^2} = 0.5$$

$$D = \sqrt{(2-9/2)^2 + (1-7/2)^2} = 3.54$$

The distance of item 3 = (4, 3) to c_1 = (3/2, 1) and with c_2 = (9/2, 7/2) is calculated as,

$$D = \sqrt{(4-3/2)^2 + (3-1)^2} = 3.20$$

$$D = \sqrt{(4-9/2)^2 + (3-7/2)^2} = 0.71$$

The distance of item 4 = (5, 4) to c_1 = (3/2, 1) and with c_2 = (9/2, 7/2) is calculated as,

$$D = \sqrt{(5-3/2)^2 + (4-1)^2} = 4.61$$

$$D = \sqrt{(5-9/2)^2 + (4-7/2)^2} = 0.71$$

Objects-centroids distance

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.3 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} c_1 = \left(\frac{3}{2}, 1\right) & \text{group 1} \\ c_2 = \left(\frac{9}{2}, \frac{7}{2}\right) & \text{group 2} \end{matrix}$$

From the above object centroid distance matrix we can see,

- Item 1 has minimum distance for group1, so we cluster item 1 in group 1.
- Item 2 has minimum distance for group 1, so we cluster item 2 in group 1.
- Item 3 has minimum distance for group 2, so we cluster item 3 in group 2.
- Item 4 has minimum distance for group 2, so we cluster item 4 in group 2.

Object Clustering

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$G^2 = G^1$, Objects does not move from group any more. So, the final clusters are as follows:

- Item 1 and 2 are clustered in group 1
- Item 3 and 4 are clustered in group 2

Ex. 6.7.4 : Suppose we have eight data points and each data point has 2 features. Cluster the data points into 3 clusters using k-means algorithm.

Data points	Attribute 1(x)	Attribute 2(y)
1	2	10
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9

☑ **Soln. :**

☞ **Initial value of centroid**

Suppose we use data points 1, 4 and 7 as the first centroids, c_1 = (2, 10), c_2 = (5, 8) and c_3 = (1, 2)

The distance of data point 1 = (2, 10) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(2-2)^2 + (10-10)^2} = 0$$

$$D = \sqrt{(2-5)^2 + (10-8)^2} = 3.61$$

$$D = \sqrt{(2-1)^2 + (10-2)^2} = 8.06$$

The distance of data point 1 = (2, 5) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(2-2)^2 + (5-10)^2} = 5$$

$$D = \sqrt{(2-5)^2 + (5-8)^2} = 4.24$$

$$D = \sqrt{(2-1)^2 + (5-2)^2} = 3.16$$

The distance of data point 1 = (8, 4) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(8-2)^2 + (4-10)^2} = 8.48$$

$$D = \sqrt{(8-5)^2 + (4-8)^2} = 5$$

$$D = \sqrt{(8-1)^2 + (4-2)^2} = 7.28$$

The distance of data point 1 = (5, 8) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(5-2)^2 + (8-10)^2} = 3.61$$

$$D = \sqrt{(5-5)^2 + (8-8)^2} = 0$$

$$D = \sqrt{(5-1)^2 + (8-2)^2} = 7.21$$

The distance of data point 1 = (7, 5) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(7-2)^2 + (5-10)^2} = 7.07$$

$$D = \sqrt{(7-5)^2 + (5-8)^2} = 3.61$$

$$D = \sqrt{(7-1)^2 + (5-2)^2} = 6.71$$

The distance of data point 1 = (6, 4) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(6-2)^2 + (4-10)^2} = 7.21$$

$$D = \sqrt{(6-5)^2 + (4-8)^2} = 4.12$$

$$D = \sqrt{(6-1)^2 + (4-2)^2} = 5.39$$

The distance of data point 1 = (1, 2) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(1-2)^2 + (2-10)^2} = 8.06$$

$$D = \sqrt{(1-5)^2 + (2-8)^2} = 7.21$$

$$D = \sqrt{(1-1)^2 + (2-2)^2} = 0$$

The distance of data point 1 = (4, 9) to c_1 = (2, 10), c_2 = (5, 8) and with c_3 = (1, 2) is,

$$D = \sqrt{(4-2)^2 + (9-10)^2} = 2.24$$

$$D = \sqrt{(4-5)^2 + (9-8)^2} = 1.4$$

$$D = \sqrt{(4-1)^2 + (9-2)^2} = 7.62$$

Objects-centroids distance

$$D^0 = \begin{bmatrix} 0 & 5 & 8.48 & 3.61 & 7.07 & 7.21 & 8.06 & 2.24 \\ 3.61 & 4.24 & 5 & 0 & 3.61 & 4.12 & 7.21 & 1.4 \\ 8.06 & 3.16 & 7.28 & 7.21 & 6.71 & 5.39 & 0 & 7.62 \end{bmatrix} \begin{matrix} c_1 = (2, 10) & \text{group 1} \\ c_2 = (5, 8) & \text{group 2} \\ c_3 = (1, 2) & \text{group 3} \end{matrix}$$

From the above object centroid distance matrix we can see,

- Data point 1 has minimum distance for group1, so we cluster data point 1 in group 1.
- Data point 2 has minimum distance for group3, so we cluster data point 2 in group 3.
- Data point 3 has minimum distance for group 2, so we cluster data point 3 in group 2.
- Data point 4 has minimum distance for group 2, so we cluster data point 4 in group 2.
- Data point 5 has minimum distance for group 2, so we cluster data point 5 in group 2.
- Data point 6 has minimum distance for group 2, so we cluster data point 6 in group 2.
- Data point 7 has minimum distance for group 3, so we cluster data point 7 in group 3.
- Data point 8 has minimum distance for group 2, so we cluster data point 8 in group 2.

Object Clustering

$$G^0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Iteration 1 : Determine centroids

C1has only one member thus $c_1 = (2, 10)$ remains same.

$$C_2 = (8 + 5 + 7 + 6 + 4/5, 4 + 8 + 5 + 4 + 9/5) = (6, 6)$$

$$C_3 = (2 + 1/2, 5 + 2/2) = (1.5, 3.5)$$

Objects-centroids distance

$$D^1 = \begin{bmatrix} 0 & 5 & 8.48 & 3.61 & 7.07 & 7.21 & 8.06 & 2.24 \\ 5.66 & 4.12 & 2.83 & 2.24 & 1.41 & 2 & 6.40 & 3.16 \\ 6.52 & 1.58 & 6.25 & 5.7 & 5.7 & 4.52 & 1.58 & 6.04 \end{bmatrix} \begin{matrix} c_1 = (2, 10) & \text{group 1} \\ c_2 = (6, 6) & \text{group 2} \\ c_3 = (1.5, 3.5) & \text{group 3} \end{matrix}$$

Object Clustering

$$G^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

☞ **Iteration 2 : Determine centroids**

$$C_1 = (2 + 4/2, 10 + 9/2) = (3, 9.5)$$

$$C_2 = (8 + 5 + 7 + 6/4, 4 + 8 + 5 + 4/4) = (6.5, 5.25)$$

$$C_3 = (2 + 1/2, 5 + 2/2) = (1.5, 3.5)$$

$$D^2 = \begin{bmatrix} 1.12 & 2.35 & 7.43 & 2.5 & 6.02 & 6.26 & 7.76 & 1.12 \\ 6.54 & 4.51 & 1.95 & 3.13 & 0.56 & 1.35 & 6.38 & 7.68 \\ 6.52 & 1.58 & 6.52 & 5.7 & 5.7 & 4.52 & 1.58 & 6.04 \end{bmatrix} \begin{matrix} c_1 = (3, 9.5) & \text{group 1} \\ c_2 = (6.5, 5.25) & \text{group 2} \\ c_3 = (1.5, 3.5) & \text{group 3} \end{matrix}$$

Object Clustering

$$G^2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Module 6

Iteration 3 : Determine centroids

$$C_1 = (2 + 5 + 4/3, 10 + 9 + 8/3) = (3.67, 9)$$

$$C_2 = (8 + 7 + 6/3, 4 + 5 + 4/3) = (7, 4.33)$$

$$C_3 = (2 + 1/2, 5 + 2/2) = (1.5, 3.5)$$

$$D^2 = \begin{bmatrix} 1.95 & 4.33 & 6.61 & 1.66 & 5.2 & 5.52 & 7.49 & 0.33 \\ 6.01 & 5.04 & 1.05 & 4.17 & 0.67 & 1.05 & 6.44 & 5.55 \\ 6.52 & 1.58 & 6.52 & 5.7 & 5.7 & 4.52 & 1.58 & 6.04 \end{bmatrix} \begin{matrix} c_1 = (3.67, 9) & \text{group 1} \\ c_2 = (7, 4.33) & \text{group 2} \\ c_3 = (1.5, 3.5) & \text{group 3} \end{matrix}$$

Object Clustering

$$G^3 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$G^3 = G^2$, Objects does not move from group any more. So, the final clusters are as follows:

- Data points 1, 4 and 8 are clustered in group 1
- Data points 3, 5 and 6 are clustered in group 2
- Data points 2 and 7 are clustered in group 3

UEx. 6.7.5 MU - May 15, May 16, 10 Marks

Apply K-means algorithm on given data for k = 3. Use c_1 (2), c_2 (16) and c_3 (38) as initial cluster centres.

Data : 2, 4, 6, 3, 31, 12, 15, 16, 38, 35, 14, 21, 23, 25, 30

Soln. :

$c_1 = 2, \quad c_2 = 16, \quad c_3 = 38$

The numbers which are close to mean are grouped into respective clusters.

$k_1 = (2, 4, 6, 3\}, \quad k_2 = \{12, 15, 16, 14, 21, 23, 25\}, \quad k_3 = \{31, 35, 30\}$

Again calculate new mean for new cluster group.

$c_1 = 3.75, \quad c_2 = 18, \quad c_3 = 32$

New clusters

$k_1 = (2, 4, 6, 3\}, \quad k_2 = \{12, 15, 16, 14, 21, 23, 25\}, \quad k_3 = \{31, 35, 30\} \quad c_1 = 3.75, \quad c_2 = 18, \quad c_3 = 32$

Clusters remains unchanged

Final clusters

$K_1 = (2, 4, 6, 3\}, \quad k_2 = \{12, 15, 16, 14, 21, 23, 25\}, \quad k_3 = \{31, 35, 30\}$

6.8 UNSUPERVISED LEARNING : HIERARCHICAL CLUSTERING

6.8.1 Hierarchical Clustering

Agglomerative Hierarchical Clustering

- In agglomerative clustering initially each data point is considered as a single cluster. In the next step, pairs of clusters are merged or agglomerated.
- This step is repeated until all clusters have been merged in to a single cluster. At the end a single cluster remains that contains all the data points.
- Hierarchical clustering algorithms works in top-down manner or bottom-up manner. Hierarchical clustering is known as Hierarchical agglomerative clustering.

- In agglomerative, clustering is represented as a dendogram as in Fig. 6.8.1 where each merge is represented by a horizontal line.

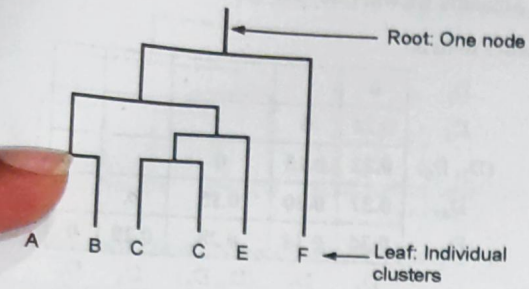


Fig. 6.8.1 : Dendogram

- A clustering of the data objects is obtained by cutting the dendogram at the desired level, then each connected forms a cluster.
- The basic steps of Agglomerative hierarchical clustering are as follows:
  1. Compute the proximity matrix (distance matrix)
  2. Assume each data point as a cluster.
  3. Repeat
  4. Merge the two nearest clusters.
  5. Update the proximity matrix

  Until only a single cluster remains
- Agglomerative hierarchical clustering proximity matrix is symmetric i.e., the number on lower half will be same as the numbers on top half.
- Different approaches to defining the distance between clusters distinguish the different algorithm's i.e., Single linkage, Complete linkage and Average linkage clusters.
- In single linkage, the distance between two clusters is considered to be equal to shortest distance from any member of one cluster to any member of other cluster.

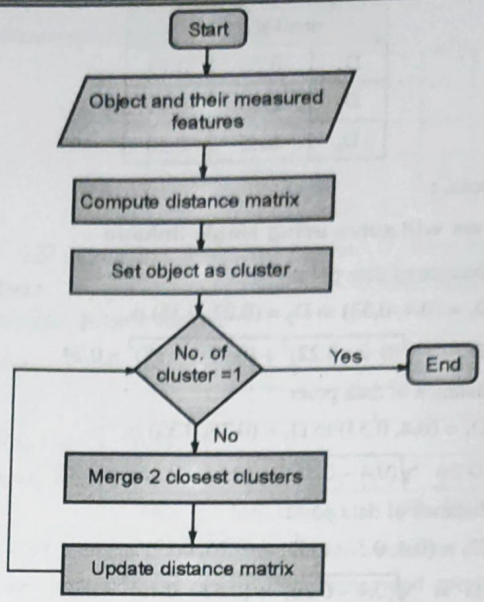


Fig. 6.8.2

D (r, s) = Min {d (i, j), object i → cluster r and object j → cluster s

- In complete linkage, the distance between two clusters is considered to be equal to greatest distance from any member of one cluster to any member of other cluster.

D (r, s) = Max {d (i, j), object i→ cluster r and object j → cluster s

- In average linkage, we consider the distance between any two clusters A and B is taken to be equal to average of all distances between pairs of object i in A and j in B.ie., mean distance between elements of each other.

D (r, s) = Mean {d (i, j), object i→ cluster r and object j → cluster s

6.8.2 Examples on Hierarchical clustering

Ex. 6.8.1 : The table shows the six data points. Use all link methods to find clusters. Use Euclidian distance measure.

	X	y
D_1	0.4	0.53
D_2	0.22	0.38
D_3	0.35	0.32

	X	y
D_4	0.26	0.19
D_5	0.08	0.41
D_6	0.45	0.30

☑ **Soln. :**

First we will solve using single linkage

The distance of data point

$D_1 = (0.4, 0.53)$ to $D_2 = (0.22, 0.38)$ is,

$$D = \sqrt{(0.4 - 0.22)^2 + (0.53 - 0.38)^2} = 0.24$$

The distance of data point

$D_1 = (0.4, 0.53)$ to $D_3 = (0.35, 0.32)$ is,

$$D = \sqrt{(0.4 - 0.35)^2 + (0.53 - 0.32)^2} = 0.22$$

The distance of data point

$D_1 = (0.4, 0.53)$ to $D_4 = (0.26, 0.19)$ is,

$$D = \sqrt{(0.4 - 0.26)^2 + (0.53 - 0.19)^2} = 0.37$$

The distance of data point

$D_1 = (0.4, 0.53)$ to $D_5 = (0.08, 0.41)$ is,

$$D = \sqrt{(0.4 - 0.08)^2 + (0.53 - 0.41)^2} = 0.34$$

The distance of data point

$D_1 = (0.4, 0.53)$ to $D_6 = (0.45, 0.30)$ is,

$$D = \sqrt{(0.4 - 0.45)^2 + (0.53 - 0.30)^2} = 0.23$$

Similarly we will calculate all distances.

☞ **Distance matrix**

	D_1	D_2	D_3	D_4	D_5	D_6
D_1	**0**					
D_2	**0.24**	**0**				
D_3	**0.22**	**0.15**	**0**			
D_4	**0.37**	**0.20**	**0.15**	**0**		
D_5	**0.34**	**0.14**	**0.28**	**0.29**	**0**	
D_6	**0.23**	**0.25**	**0.11**	**0.22**	**0.39**	**0**

0.11 is smallest. D_3 and D_6 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((D_3, D_6), D_1)$ = min (distance (D_3, D_1), distance (D_6, D_1)) = min (0.22, 0.23) = 0.22

Distance $((D_3, D_6), D_2)$ = min (distance (D_3, D_2), distance (D_6, D_2)) = min (0.15, 0.25) = 0.15

Distance $((D_3, D_6), D_4)$ = min (distance (D_3, D_4), distance (D_6, D_4)) = min (0.15, 0.22) = 0.15

Distance $((D_3, D_6), D_5)$ = min (distance (D_3, D_5), distance (D_6, D_5)) = min (0.28, 0.39) = 0.28

Similarly we will calculate all distances.

Distance matrix

	D_1	D_2	(D_3, D_6)	D_4	D_5
D_1	**0**				
D_2	**0.24**	**0**			
(D_3, D_6)	**0.22**	**0.15**	**0**		
D_4	**0.37**	**0.20**	**0.15**	**0**	
D_5	**0.34**	**0.14**	**0.28**	**0.29**	**0**

0.14 is smallest. D_2 and D_5 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((D_3, D_6), (D_2, D_5))$ = min (distance (D_3, D_2), distance (D_6, D_2), distance (D_3, D_5), distance (D_6, D_6))

= min (0.15, 0.25, 0.28, 0.29) = 0.15

Similarly, we will calculate all distances.

Distance matrix

	D_1	(D_2, D_5)	(D_3, D_6)	D_4
D_1	**0**			
(D_2, D_5)	**0.24**	**0**		
(D_3, D_6)	**0.22**	**0.15**	**0**	
D_4	**0.37**	**0.20**	**0.15**	**0**

0.15 is smallest. (D_2, D_5) and (D_3, D_6) as well as D_4 and (D_3, D_6) have smallest distance. We can pick either one.

Distance matrix

	D_1	(D_2, D_5, D_3, D_6)	D_4
D_1	**0**		
(D_2, D_5, D_3, D_6)	**0.22**	**0**	
D_4	**0.37**	**0.15**	**0**

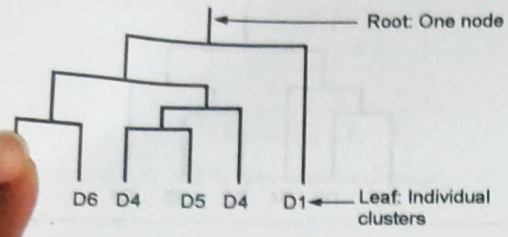
0.15 is smallest. (D_2, D_5, D_3, D_6) and D_4 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

	D_1	$(D_2, D_5, D_3, D_6, D_4)$
D_1	**0**	
$(D_2, D_5, D_3, D_6, D_4)$	**0.22**	**0**

Now a single cluster remains (D_2, D_5, D_3, D_6, D_4, D_1)

Next, we represent the final dendogram for single linkage as,



Now we will solve using complete linkage

Distance matrix

	D_1	D_2	D_3	D_4	D_5	D_6
D_1	0					
D_2	0.24	0				
D_3	0.22	0.15	0			
D_4	0.37	0.20	0.15	0		
D_5	0.34	0.14	0.28	0.29	0	
D_6	0.23	0.25	0.11	0.22	0.39	0

0.11 is smallest. D_3 and D_6 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((D_3, D_6), D_1)$ = max (distance (D_3, D_1), distance (D_6, D_1)) = max (0.22, 0.23) = 0.23

Similarly, we will calculate all distances.

Distance matrix

	D_1	D_2	(D_3, D_6)	D_4	D_5
D_1	**0**				
D_2	**0.24**	**0**			
(D_3, D_6)	**0.23**	**0.25**	**0**		
D_4	**0.37**	**0.20**	**0.22**	**0**	
D_5	**0.34**	**0.14**	**0.39**	**0.29**	**0**

0.14 is smallest. D_2 and D_5 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

	D_1	(D_2, D_5)	(D_3, D_6)	D_4
D_1	0			
(D_2, D_5)	0.34	0		
(D_3, D_6)	0.23	0.39	0	
D_4	0.37	0.29	0.22	0

0.22 is smallest. Here (D_3, D_6) and D_4 have smallest distance. So, we combine these two in one cluster and recalculate distance matrix.

Distance matrix

	D_1	(D_3, D_6, D_4)	(D_3, D_6, D_4)
D_1	0		
(D_2, D_5)	0.34	0	
(D_3, D_6, D_4)	0.37	0.39	0

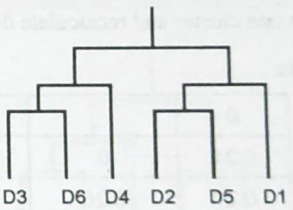
0.34 is smallest. (D_2, D_5) and D_1 have smallest distance so, we combine these two in one cluster and recalculate distance matrix.

Distance matrix

	(D_2, D_5, D_1)	(D_3, D_6, D_4)
(D_2, D_5, D_1)	**0**	**0**
(D_3, D_6, D_4)	**0.39**	**0**

Now a single cluster remains (D_2, D_5, D_1, D_3, D_6, D_4)

Next, we represent the final dendogram for complete linkage as,



Now we will solve using average linkage

Distance matrix

	D_1	D_2	D_3	D_4	D_5	D_6
D_1	0					
D_2	0.24	0				
D_3	0.22	0.15	0			
D_4	0.37	0.20	0.15	0		
D_5	0.34	0.14	0.28	0.29	0	
D_6	0.23	0.25	0.11	0.22	0.39	0

0.11 is smallest. D_3 and D_6 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((D_3, D_6), D_1) = 1/2$ (distance (D_3, D_1) + distance $(D_6, D_1)) = 1/2\ (0.22 + 0.23) = 0.23$

Similarly, we will calculate all distances.

Distance matrix

D_1	0				
D_2	0.24	0			
(D_3, D_6)	0.23	0.2	0		
D_4	0.37	0.20	0.19	0	
D_5	0.34	0.14	0.34	0.29	0
	D_1	D_2	(D_3, D_6)	D_4	D_5

0.14 is smallest. D_2 and D_5 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

D_1	0			
(D_2, D_5)	0.29	0		
(D_3, D_6)	0.22	0.27	0	
D_4	0.37	0.22	0.15	0
	D_1	(D_2, D_5)	(D_3, D_6)	D_4

(D_3, D_6) and D_4 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

D_1	0		
(D_2, D_5)	0.24	0	
(D_3, D_6, D_4)	0.27	0.26	0
	D_1	(D_2, D_5)	(D_3, D_6, D_4)

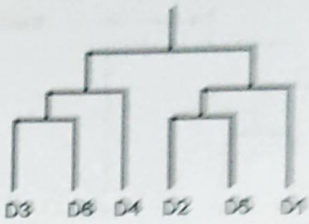
0.24 is smallest. (D_2, D_5) and D_1 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

(D_2, D_5, D_1)	**0**	**0**
(D_3, D_6, D_4)	**0.26**	**0**
	(D_2, D_5, D_1)	(D_3, D_6, D_4)

Now a single cluster remains $(D_2, D_5, D_1, D_3, D_6, D_4)$

Next, we represent the final dendogram for average linkage as,



Ex. 6.8.2 : Apply single linkage, complete linkage and average linkage on the following distance matrix and draw dendogram.

☑ **Soln. :**

First we will solve using single linkage

Distance matrix

P_1	0				
P_2	2	0			
P_3	6	3	0		
P_4	10	9	7	0	
P_5	9	8	5	4	0
	P_1	P_2	P_3	P_4	P_5

2 is smallest. P_1 and P_2 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((P_1, P_2), P_3) = \min$ (distance (P_1, P_3), distance $(P_2, P_3)) = \min\ (6, 3) = 3$

Similarly, we will calculate all distances.

Distance matrix

(P_1, P_2)	0			
P_3	3	0		
P_4	9	7	0	
P_5	8	5	4	0
	(P_1, P_2)	P_3	P_4	P_5

3 is smallest. (P_1, P_2) and P_3 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((P_1, P_2, P_3), P_4)) = \min$ (distance (P_1, P_4), distance (P_2, P_4), distance $(P_3, P_4)) = \min\ (9, 7) = 7$

Similarly, we will calculate all distances.

Distance matrix

(P_1, P_2, P_3)	0		
P_4	7	0	
P_5	5	4	0
	(P_1, P_2, P_3)	P_4	P_5

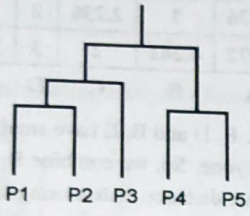
4 is smallest. P_4 and P_5 have smallest distance.

Distance matrix

(P_1, P_2, P_3)	0	
(P_4, P_5)	5	0
	(P_1, P_2, P_3)	(P_4, P_5)

Now a single cluster remains $(P_1, P_2, P_3, P_4, P_5)$

Next, we represent the final dendogram for single linkage as,



Now we will solve using complete linkage

Distance matrix

P_1	0				
P_2	2	0			
P_3	6	3	0		
P_4	10	9	7	0	
P_5	9	8	5	4	0
	P_1	P_2	P_3	P_4	P_5

2 is smallest. P_1 and P_2 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((P_1, P_2), P_3)$ = max (distance (P_1, P_3), distance (P_2, P_3)) = max (6, 3) = 6

Similarly, we will calculate all distances.

Distance matrix

(P_1, P_2)	0			
P_3	6	0		
P_4	10	7	0	
P_5	9	5	4	0
	(P_1, P_2)	P_3	P_4	P_5

4 is smallest. P_4 and P_5 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

(P_1, P_2)	0		
P_3	6	0	
(P_4, P_5)	10	7	0
	(P_1, P_2)	P_3	(P_4, P_5)

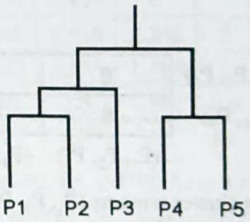
6 is smallest. (P_1, P_2) and P_3 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

(P_1, P_2, P_3)	0	
(P_4, P_5)	10	0
	(P_1, P_2, P_3)	(P_4, P_5)

Now a single cluster remains $(P_1, P_2, P_3, P_4, P_5)$

Next, we represent the final dendogram for complete linkage as,



Now we will solve using average linkage

Distance matrix

P_1	0				
P_2	2	0			
P_3	6	3	0		
P_4	10	9	7	0	
P_5	9	8	5	4	0
	P_1	P_2	P_3	P_4	P_5

Module 6

2 is smallest. P_1 and P_2 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance $((P_1, P_2), P_3)$ = 1/2 (distance (P_1, P_3), distance (P_2, P_3)) = 1/2 (6, 3) = 4.5

Similarly, we will calculate all distances.

Distance matrix

(P_1, P_2)	0			
P_3	4.5	0		
P_4	9.5	7	0	
P_5	8.5	5	4	0
	(P_1, P_2)	P_3	P_4	P_5

4 is smallest. P_4 and P_5 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

(P_1, P_2)	0		
P_3	4.5	0	
(P_4, P_5)	9	6	0
	(P_1, P_2)	P_3	(P_4, P_5)

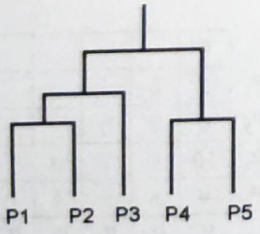
4.5 is smallest. (P_1, P_2) and P_3 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

(P_1, P_2, P_3)	0	
(P_4, P_5)	8	0
	(P_1, P_2, P_3)	(P_4, P_5)

Now a single cluster remains $(P_1, P_2, P_3, P_4, P_5)$

Next, we represent the final dendogram for average linkage as,



UEx. 6.8.3 MU - May 16, 10 Marks

Apply Agglomerative clustering algorithm on given data and draw dendrogram. Show three clusters with its allocated points. Use single link method.

	A	B	C	D	E	F
A	0	$\sqrt{2}$	$\sqrt{10}$	$\sqrt{17}$	$\sqrt{5}$	$\sqrt{20}$
B	$\sqrt{2}$	0	$\sqrt{8}$	3	1	$\sqrt{18}$
C	$\sqrt{10}$	$\sqrt{8}$	0	$\sqrt{5}$	$\sqrt{5}$	2
D	$\sqrt{17}$	1	$\sqrt{5}$	0	2	3
E	$\sqrt{5}$	1	$\sqrt{5}$	2	0	$\sqrt{13}$
F	$\sqrt{20}$	$\sqrt{18}$	2	3	$\sqrt{13}$	0

☑ **Soln. :**

Distance matrix

A	0					
B	1.414	0				
C	3.162	2.828	0			
D	4.123	1	2.236	0		
E	2.236	1	2.236	2	0	
F	4.472	4.242	2	3	3.6	0
	A	B	C	D	E	F

1 is smallest. B, D and B, E have smallest distance. We can select anyone. So, we combine B, D in one cluster and recalculate distance matrix using single linkage.

Distance matrix

A	0				
B,D	1.414	0			
C	3.162	2.236	0		
E	2.26	1	2.236	0	
F	4.472	3	2	3.6	0
	A	B,D	C	E	F

1 is smallest. B, D and E have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

A	0			
B,D,E	1.414	0		
C	3.162	1	0	
F	4.472	3	2	0
	A	B,D,E	C	F

1 is smallest. B, D, E and C are combined together.

tance matrix

A	0		
B,D,E,C	1.414	0	
F	4.472	2	0
	A	B,D,E,C	F

In the questions three clusters are asked with their allocated points. Three clusters are A, (B, D, E, C) and F.

UEx. 6.8.4 **MU - May 17, 10 Marks**

For the given set of points identify clusters using complete link and average link using Agglomerative clustering.

	A	B
P_1	1	1
P_2	1.5	1.5
P_3	5	5
P_4	3	4
P_5	4	4
P_6	3	3.5

☑ **Soln. :**

First we will solve using complete linkage

Distance matrix

p1	0					
P2	0.707	0				
P3	5.656	4.949	0			
P4	3.605	2.915	2.236	0		
P5	4.242	3.535	1.414	1	0	
P6	5.201	2.5	1.802	0.5	1.118	0
	P1	P2	P3	P4	P5	P6

0.5 is smallest. P4 and P6 have smallest distance. We can select anyone. So, we combine this in one cluster and recalculate distance matrix using complete linkage.

Distance matrix

P1	0				
P2	0.707	0			
P3	5.656	4.949	0		
P4,P6	5.201	2.5	1.802	0	
P5	4.242	3.535	1.414	1.118	0
	P1	P2	P3	P4,P6	P5

0.707 is smallest. P1 and P2 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

P1,P2	0			
P3	5.656	0		
P4,P6	5.201	2.236	0	
P5	4.242	1.414	1.118	0
	P1,P2	P3	P4,P6	P5

1.118 is smallest. P4, P6 and P5 are combined together.

Distance matrix

P1,P2	0		
P3	5.656	0	
P4,P5,P6	5.201	2.236	0
	P1,P2	P3	P4,P5,P6

2.236 is smallest. P4, P5, P6 and P3 are combined together.

P1,P2	0	
P3,P4,P5,P6	5.656	0
	P1,P2	P3,P4,P5,P6

Next we will combine all clusters in a single cluster.

Now we will solve using average linkage

Distance matrix

P1	0					
P2	0.707	0				
P3	5.656	4.949	0			
P4	3.605	2.915	2.236	0		
P5	4.242	3.535	1.414	1	0	
P6	5.201	2.5	1.802	0.5	1.118	0
	P1	P2	P3	P4	P5	P6

0.5 is smallest. P4 and P6 have smallest distance. We can select anyone .So, we combine this in one cluster and recalculate distance matrix using complete linkage.

Distance matrix

P1	0				
P2	0.707	0			
P3	5.656	4.949	0		
P4,P6	4.403	2.707	2.019	0	
P5	4.242	3.535	1.414	1.059	0
	P1	P2	P3	P4,P6	P5

Module 6

0.707 is smallest. P1 and P2 have smallest distance. So, we combine this two in one cluster and recalculate distance matrix.

Distance matrix

P1,P2	0			
P3	5.302	0		
P4,P6	3.55	2.019	0	
P5	3.888	1.414	1.059	0
	P1, P2	P3	P4, P6	P5

1.059 is smallest. P4, P6 and P5 are combined together.

Distance matrix

P1,P2	0		
P3	5.302	0	
P4,P5,P6	3.66	1.817	0
	P1,P2	P3	P4,P5,P6

1.817 is smallest. P4,P5,P6 and P3 are combined together.

P1,P2	0	
P3,P4,P5,P6	4.07	0
	P1,P2	P3,P4,P5,P6

Next we will combine all clusters in a single cluster.

6.9 UNSUPERVISED LEARNING : ASSOCIATION RULES

6.9.1 Introduction to Association Rules

- Associations are any things that are generally occurred together. Let us understand this with the help of an example, as a shop owner we are interested in finding out the products that are generally sell together.
- We will simplify this further, if there is a customer who purchases milk then it is more likely that he will also purchase bread. From this we can say that the customers who buy milk tend to also buy bread.
- In a bakery shop most clients will buy cake. This means that there will be many frequent item sets involving cake, such as {candle, cake}.
- This might suggest the construction of an association rule **if candle then cake** – however, this is predictable given that {cake} is already a frequent item set (and clearly at least as frequent as {candle, cake}).
- Of more interest would be the converse rule **if cake then candle** which expresses that a considerable proportion of the people buying cake also buy a candle.

6.9.2 Apriori Algorithm

- Frequent item sets are used to create association rules in Apriori algorithm. It is created to work on the databases that contain transactions. Association rules are used to identify whether the two items are strongly connected to each other or not.
- Breadth first search and hash tree methods are used to find out the item sets in an optimum manner. Iterations are applied to find the frequent item sets from the huge amount of database.
- In market basket analysis mostly Apriori algorithm is used. The algorithm is used to find the products that can be bought together. Apriori algorithm can also be used in the medical discipline to find reactions of drugs on the patients.

What is Frequent Item set?

- Item sets having support more than the set minimum support or threshold value are called as Frequent item sets. Let us see an example, if there are P and Q frequent item sets, then individually P and Q should also be the frequent item set.
- Let us assume there are the two transactions: P = {3,4,6,7,8}, and Q = {4,6,9}, in these two transactions, 4 and 6 are the frequent item sets.

Apriori Algorithm Working

Apriori algorithm works according to the following steps,

1. Find the support of item sets in the transactional database. Choose the minimum support and confidence.
2. From the transaction consider all supports with more support value than the minimum or chosen support value.
3. Identify all the rules of these subsets which are having more confidence value than the threshold or minimum confidence.
4. Sort the rules as the decreasing order of lift.

- **Example :** Let us assume that we are having following dataset that contains various transactions. We want to identify the frequent item sets and create the

association rules with Apriori algorithm. In these example minimum support 2 and minimum confidence 50% is given.

Transaction ID	Item sets
T1	1,2
T2	2,4
T3	2,3
T4	1,2,4
T5	1,3
T6	2,3
T7	1,3
T8	1,2,3,5
T9	1,2,3

☑ **Soln. :**

▶ **Step 1 : Calculating S1 and I1 :**

- Initially we will generate a table that contains support count of each item set in the given dataset. This table is called the Candidate set or S1.

Itemset	Support_count
1	6
2	7
3	5
4	2
5	1

- Next, we will consider all the item sets that are having more support count than the Minimum Support i.e. 2. This will give us the table for the frequent item set I1.
- Here all the item sets are having more or same support count than the minimum support, except the E, so E item set will be removed.

Itemset	Support_count
1	6
2	7
3	5
4	2

▶ **Step 2 : Calculating S2 and I2**

- Next, we will create S2 using I1. In S2, we will generate the pair of the item sets of I1 in the form of subsets.
- Once we create the subsets, we will again find the support count from the main transaction table. Here we will check how many times these pairs have occurred together in the given dataset. So, we will get the following table for S2:

Itemset	Support_count
{1,2}	4
{1,3}	4
{1,4}	1
{2,3}	4
{2,4}	2
{3,5}	0

- Now we will compare S2 Support count with the minimum support count, and after doing this, the item set with less support count will be removed from the table S2. It will give us the following table for I2

Itemset	Support_count
{1,2}	4
{1,3}	4
{2,3}	4
{2,4}	2

▶ **Step 3 : Calculation of S3, and I3:**

- For S3, we will follow the same two processes, but now we will form the S3 table with subsets of three item sets together, and will calculate the support count from the dataset. It will give the following table:

Itemset	Support_count
{1,2,3}	2
{2,3,4}	1
{1,3,4}	0
{1,2,4}	0

- Next we will generate the I3 table. From S3 table we can see that there is only one combination of itemset that has support count same as that of minimum support count. So, the I3 will have only one combination, i.e., {1,2,3}.

Step 4 : Identifying the association rules for the subsets:

- To create the association rules,, we will create a new table with the possible rules from the above combination {1,2,3}.
- We will compute the Confidence using formula sup (1 ^ 2)/ 1 for all the rules.
- After computing the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).
- Consider the below table :

Rules	Support	Confidence
1 ^2 → 3	2	Sup{(1 ^2) ^3}/sup(1 ^2) = 2/4=0.5=50%
2^3 → 1	2	Sup{(2^3) ^1}/sup(2 ^3) = 2/4=0.5=50%
1^3 → 2	2	Sup{(1 ^3) ^2}/sup(1 ^3) = 2/4=0.5=50%
3→ 1 ^2	2	Sup{(3^(1 ^2)}/sup(3) = 2/5=0.4=40%
1→ 2^3	2	Sup{(1^(2 ^3)}/sup(1) = 2/6=0.33=33.33%
2→ 2^3	2	Sup{(2^(2 ^3)}/sup(2) = 2/7=0.28=28%

- As the given threshold or minimum confidence is 50%, so the first three rules 1^ 2 → 3, 2 ^ 3 → 1, and 1 ^ 3 → 2 can be considered as the strong association rules for the given problem.

6.9.3 Performance Measures

- Association rule mining is a kind of unsupervised learning which checks the dependency of one data item on other data item.
- Based on dependency items are mapped to each other so that it can be more profitable.
- This method identifies some interesting relations or associations that exist between the variables of dataset. For Association rule mining Apriori algorithm can be used.
- Association rule mining works on the concept of If and Else Statement, such as if X then Y. In this the If part is known as *antecedent*, and then part is known as *Consequent*.
- Such types of relationships in which we can find out some association or relation among two items is called as ***single cardinality***. In this rules are created. Here if the number of items are more then cardinality also increases accordingly.
- Due to this to measure the associations between more numbers of data items, several metrics are used. These metrics are explained bellow,

1. Support

Support is defined as the number of occurrences of an item in the dataset. It is also defined as the fraction of the transaction T that contains the itemset I.. If there are I datasets, then for transactions T, it can be written as:

Support (I) = Freq (I) / T

2. Confidence

Confidence represents how likely the rule has been found to be true. We can also say that how many times the items P and Q appears together in the dataset when the occurrence of P is already given. It is the ratio of the transaction that contains P and Q to the number of records that contain P.

Confidence = Freq (P, Q) / Freq (P)

3. Lift

Lift is defined as the strength of any rule.

Lift = Support (P, Q) / (Support (P) * Support (Q))

It is the ratio of the observed support measure and expected support if P and Q are independent of each other. It has three possible values:

- Lift= 1: It defines the probability of appearance of antecedent and consequent independent to each other.
- Lift>1: It defines the degree by which the two item sets are dependent on each other.
- Lift<1: It describes if there are any alternative for other items.

6.10 ISSUES IN MACHINE LEARNING

UQ. What are the issues in Machine learning ? (MU - May 15, 5 Marks)

1. Which algorithm we have to select to learn general target functions from specific training dataset? What should be the settings for particular algorithms, so as to converge to the desired function, given sufficient training data? Which algorithms perform best for which type of problems and representations?
2. How much training data is sufficient? What should be the general amount of data that can be found to relate the confidence in learned hypotheses to the amount training experience and the character of the learner's hypothesis space?
3. Prior knowledge held by the learner is used at which time and manner to guide the process of generalizing from examples? If we have approximately correct knowledge, will it helpful even when it is only approximately correct?
4. What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy after the complexity of the learning problem?
5. To reduce the task of learning to one or more function approximation problems, what will be the best approach? What specific functions should the system attempt to learn? Can this process itself be automated?
6. To improve the knowledge representation and to learn the target function, how can the learner automatically alter its representation?

6.11 HOW TO CHOOSE THE RIGHT ALGORITHM?

UQ. Explain the steps required for selecting the right machine learning algorithm. (MU - May 16, 8 Marks)

With all the different algorithms available in machine learning, how can you select which one to use ? First you need to focus on your goal. What are you trying to get out of this? What data do you have or can you collect? Secondly you have to consider the data.

1. **Goal :** If you are trying to predict or forecast a target value, then you need to look into supervised learning. Otherwise, you have to use unsupervised learning.
   (a) If you have chosen **supervised** learning, then next you need to focus on what's your **target** value?
   If **target value** is **discrete** (e.g. Yes/ No, 1 /2/3, A/B/C), then use **Classification**.
   If **target value** is continuous i.e. Number of values (e.g. 0 – 100, – 99 to 99), then use **Regression.**
   (b) If you have chosen **unsupervised** learning, then next you need to focus on what is your **aim?**
   If you want to **fit your data** into some **discrete groups,** then use **Clustering**
   If you want to **find numerical estimate** of how strong the fit into each group, then use **density estimation algorithm**
2. **Data :** Are the features continuous or nominal ? Are there missing values in features ? If yes, what is a reason for missing values? Are there outliers in the data? To narrow the algorithm selection process, all of these features of your data can help you.

Table 6.11.1 : Selection of Algorithm

	Supervised Learning	Unsupervised Learning
Discrete	Classification	Clustering
Continuous	Regression	Density Estimation

6.12 STEPS IN DEVELOPING A MACHINE LEARNING APPLICATION

UQ. Explain the steps of developing Machine Learning applications. (MU - May 19, 10 Marks)

1. Collection of Data

You could collect the samples from a website and extracting data.

- From RSS feed or an API
- From device to collect wind speed measurement
- Publicly available data.

Module 6

2. Preparation of the input data

- Once you have the input data, you need to check whether it's in a useable format or not.
- Some algorithm can accept target variables and features as string; some need them to be integers.Some algorithm accepts features in a special format.

3. Analyse the input data

- Looking at the data you have passed in a text editor to check collection and preparation of input data steps are properly working and you don't have a bunch of empty values.
- You can also check at the data to find out if you can see any patterns or if there is anything obvious, such as a few data points greatly differ from remaining set of the data.Plotting data in 1, 2 or 3 dimensions can also help.
- Distil multiple dimensions down to 2/3 so that you can visualize the data.

4. The importance of this step is that it makes you understand that you don't have any garbage value coming in.

5. Train the algorithm

- Good clean data from the first two steps is given as input to the algorithm. The algorithm extracts information or knowledge. This knowledge is mostly stored in a format that is readily useable by machine for next 2 steps.
- In case of unsupervised learning, training step is not there because target value is not present. Complete data is used in the next step.

6. Test the algorithm

- In this step the information learned in the previous step is used. When you are checking an algorithm, you will test it to find out whether it works properly or not. In supervised case, you have some known values that can be used to evaluate the algorithm.
- In case of unsupervised, you may have to use some other matrices to evaluate the success. In either case, if you are not satisfied, you can again go back to step 4, change some things and test again.
- Mostly problem occurs in collection or preparation of data and you will have to go back to step 1.

7. Use it

In this step a real program is developed to do some task, and once again it is checked if all the previous steps worked as you expected. You might encounter some new data and have to revisit step 1-5.

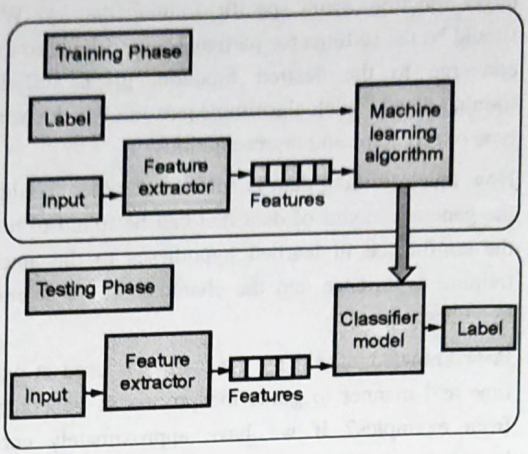


Fig. 6.12.1 : Typical example of Machine Learning Application

6.13 APPLICATIONS OF MACHINE LEARNING

UQ. Write short note on : Machine learning applications. **(MU - May 16, May 17, 10 Marks)**

(1) Learning Associations

- A supermarket chain-one an example of retail application of machine learning is basket analysis, which is finding associations between products bought by customers :
- If people who buy P typically also buy Q and if there is a customer who buys Q and does not buy P, he or she is a potential P customer. Once we identify such customers, we can target them for cross-selling.
- In finding an association rule, we are interested in learning a conditional probability of the form P (Q/P) where Q is the product we would like to condition on P, which are the product / products which we know that customer has already purchased.

P (Milk / Bread) = 0.7

- It implies that 70% of customers who buy bread also buy milk

(2) Classification

- A credit is an amount of money loaned by a financial institution.
- It is important for the bank to be able to predict in advance the risk associated with a loan. Which is the probability that the customer will default and not pay the whole amount back?
- In credit scoring, the bank calculates the risk given the amount of credit and the information about the customer. (Income, savings, collaterals, profession, age, past financial history). The aim is to infer a general rule from this data, coding the association between a customer's attributes and his risk.
- Machine Learning system fits a model to the past data to be able to calculate the risk for a new application and then decides to accept or refuse it accordingly.

  If income > Q_1 and savings > Q_2

  Then low – risk ELES high – risk
- Other classification examples are Optical character recognition, face recognition, medical diagnosis, speech recognition and biometric.

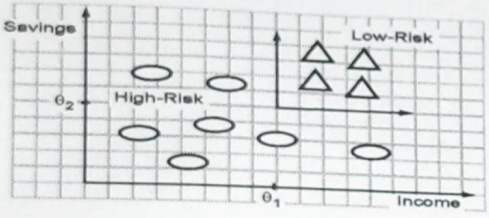


Fig. 6.13.1 : Classification for credit scoring

(3) Regression

- Suppose we want to design a system that can predict the price of a flat.
- Let's take the inputs as the area of the flat, location and purchase year and other information that affects the rate of flat.
- The output is the price of the flat. The applications where output is numeric are regression problems.
- Let X represents flat features and Y is the price of flat. We can collect training data by surveying past purchased transactions and the Machine Learning algorithm fits a function to this data to learn Y as a function of X for the suitable values of W and W_0.

$$Y = w^*x + w_0$$

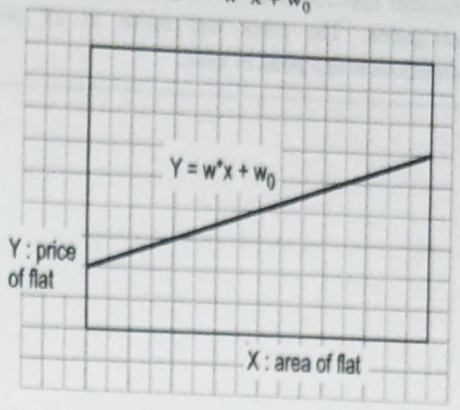


Fig. 6.13.2 : Regression for prediction of price of flat

(4) Unsupervised Learning

- One of the important unsupervised learning problem is clustering. In clustering dataset is partitioned in to meaningful sub classes known as clusters. For example, suppose you want to decorate your home using given items.
- Now you will classify them using unsupervised learning (no prior knowledge) and this classification can be on the basis of color of items, shape of items, material used for items, type of items or whatever way you would like.

(5) Reinforcement Learning

- There are some of the applications where output of system is a sequence of actions. In such applications the sequence of correct actions instead of single action is important in order to reach goal.
- An action is said to be good if it is part of good policy. Machine learning program generates a policy by learning previous good action sequences. Such methods are called reinforcement methods
- A good example of reinforcement learning is chess playing. In artificial intelligence and machine learning, one of the most important research area is game playing.

Module 6

- Games can be are easily described but at the same time, they are quite difficult to play well. Let's take a example of chess that has limited number of rules, but the game is very difficult because for each state there can be large number of possible moves.
- Another application of reinforcement learning is robot navigation. The robot can move in all possible directions at any point of time.
- The algorithm should reach goal state from an initial state by learning the correct sequence of actions after conducting number of trial runs.
- When the system has unreliable and partial sensory information, it makes reinforcement learning complex. Let's take an example of robot with incomplete camera information. Here robot does not know its exact location.

Chapter Ends...

❑❑❑