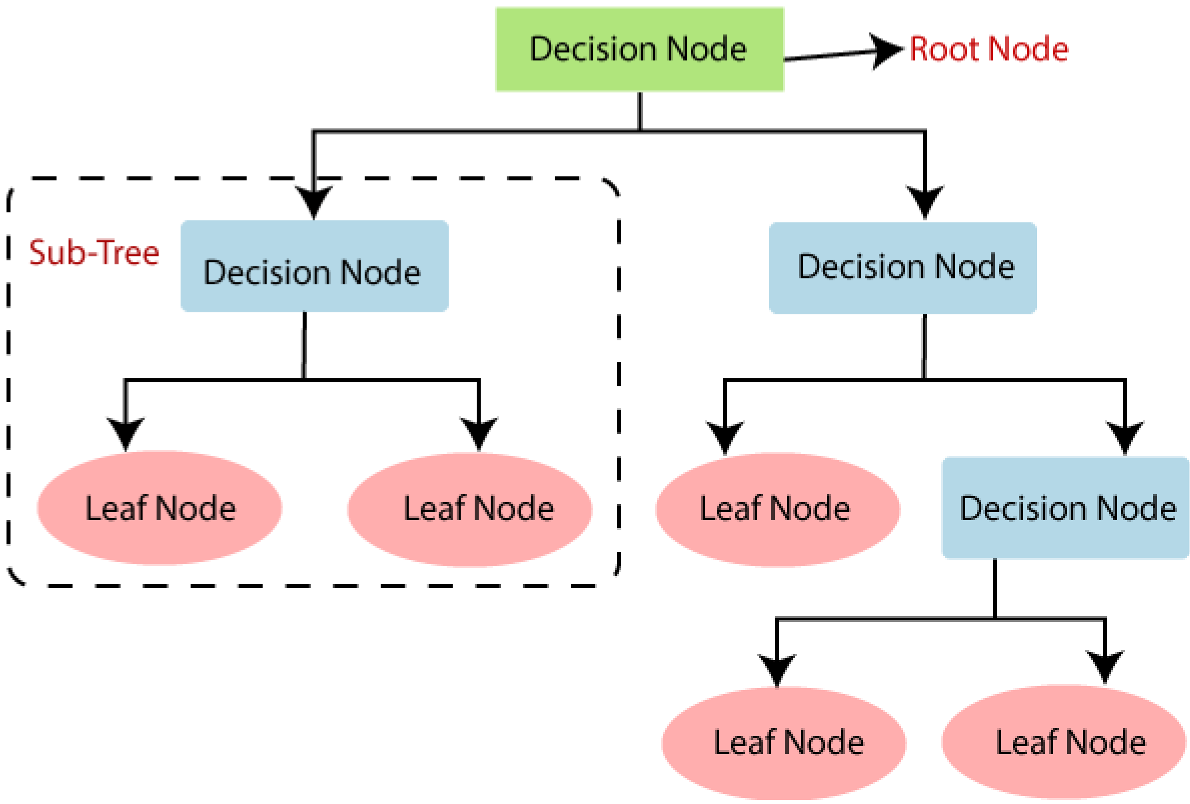
Chapter No.3 Basic Classification

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Learning with Trees: Decision Tree Classification Algorithm

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

Note: A decision tree can contain categorical data (YES/NO) as well as numeric data.



Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

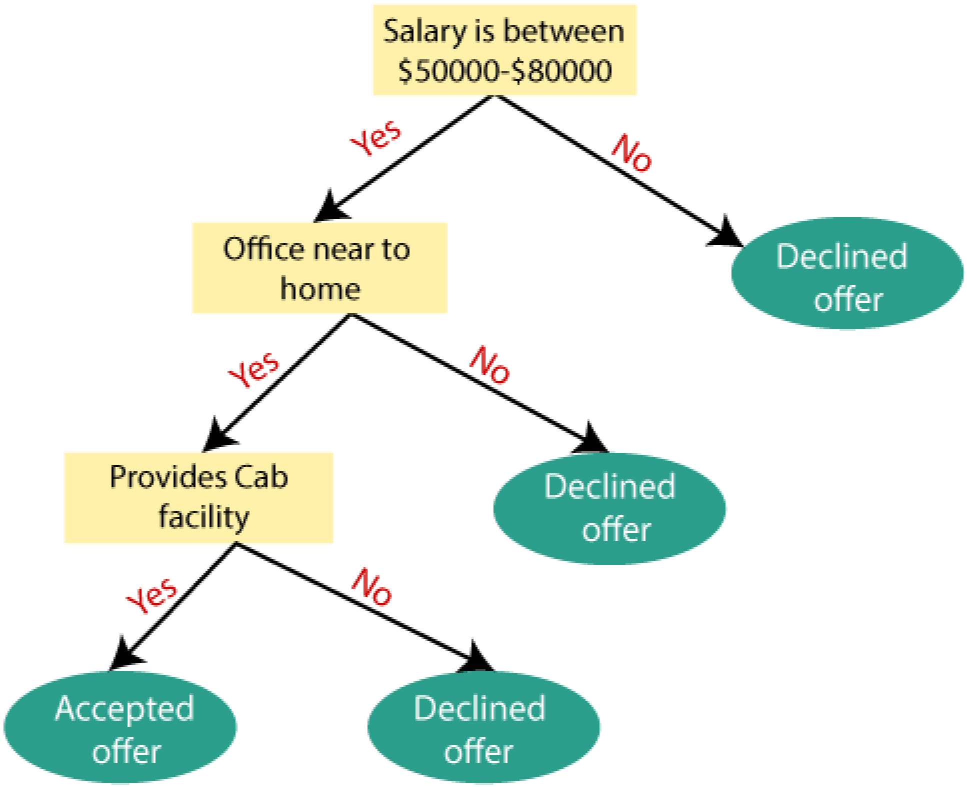
How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- **Information Gain**
- **Gini Index**

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.

- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

1. Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)]

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

Where,

- **S= Total number of samples**
- **P(yes)= probability of yes**
- **P(no)= probability of no**

2. Gini Index:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

Gini Index= 1- $\sum_j P_j^2$

Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree **pruning** technology used:

- **Cost Complexity Pruning**
- **Reduced Error Pruning.**

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
- For more class labels, the computational complexity of the decision tree may

Steps for decision trees algorithm implementation which are given below:

- **Data Pre-processing step**
- **Fitting a Decision-Tree algorithm to the Training set**
- **Predicting the test result**
- **Test accuracy of the result(Creation of Confusion matrix)**
- **Visualizing the test set result.**

Decision trees are often used while implementing machine learning algorithms. The hierarchical structure of a decision tree leads us to the final outcome by traversing through the nodes of the tree. Each node consists of an attribute or feature which is further split into more nodes as we move down the tree. But how do we decide:

- *Which attribute/feature should be placed at the root node?*
- *Which features will act as internal nodes or leaf nodes?*

To decide this, and how to split the tree, we use splitting measures like Gini Index, Information Gain, etc. We will learn how the Gini Index can be used to split a decision tree.

Before starting with the Gini Index, let us first understand what splitting is and what are the measures used to perform it.

What are Splitting Measures?

With more than one attribute taking part in the decision-making process, it is necessary to decide the relevance and importance of each of the attributes. Thus placing the most relevant at the root node and further traversing down by splitting the nodes.

As we move further down the tree, the level of impurity or uncertainty decreases, thus leading to a better classification or best split at every node. To decide the same, splitting measures such as Information Gain, Gini Index, etc. are used.

What is Information Gain?

Information Gain is used to determine which feature/attribute gives us the maximum information about a class.

- Information Gain is based on the concept of entropy, which is the degree of uncertainty, impurity or disorder.

- Information Gain aims to reduce the level of entropy starting from the root node to the leave nodes.

Formula for Entropy

$$\text{Entropy} = -\sum_{i=1}^{n} p_i * \text{Log}_2(p_i)$$

Entropy Formula

Here “p” denotes the probability that it is a function of entropy.

Entropy is not preferred due to the ‘log’ function as it increases the computational complexity.

What is Gini Index?

Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

But what is actually meant by ‘impurity’?
If all the elements belong to a single class, then it can be called pure. The degree of Gini index varies between 0 and 1,
where,
0 denotes that all elements belong to a certain class or if there exists only one class, and
1 denotes that the elements are randomly distributed across various classes.

A Gini Index of 0.5 denotes equally distributed elements into some classes.

Gini Index Formula

$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

Gini Index Formula

Where Pi denotes the probability of an element being classified for a distinct class.

While building the decision tree, we would prefer choosing the attribute/feature with the least Gini index as the root node.

Example No. 1 Design the decision Tree using Gini Index for following Data.

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Solution:

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Compute the Gini Index for the overall collection of training examples.
- There are four possible output variables Cinema, Tennis, Stay In and Shopping.
- The data has 6 instances of Cinema, 2 instances of Tennis, 1 instance of Stay In and 1 of shopping.
- $Gini(S) = 1 - [\left(\frac{6}{10}\right)^2 + \left(\frac{2}{10}\right)^2 + \left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^2] = 0.58$

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Computation of **Gini Index for Money** Attribute
- It has **two possible values of Rich (7 examples)** and **Poor (3 examples).**
- For **Money = Poor,** there are **3 examples with "Cinema".**
- $Gini(S) = 1 - [(\frac{3}{3})^2] = 0$ ✓ 7
- For **Money = Rich,** there are **2 examples with "Tennis", 3 examples with "Cinema"** and **1 example with "Stay in", "Shopping" each**
- $Gini(S) = 1 - [(\frac{2}{7})^2 + (\frac{3}{7})^2 + (\frac{1}{7})^2 + (\frac{1}{7})^2] =$ 0.694
- $\boldsymbol{Weighted\ Average(Money)}$

$$= \mathbf{0 * \left(\frac{3}{10}\right) + 0.694 * \left(\frac{7}{10}\right) = 0.486}$$

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Computation of **Gini Index for Parents** Attribute
- It has two possible values of **Yes (5 examples)** and **No (5 examples).**
- For **Parents = Yes,** there are **5 examples**, all with "Cinema".
- $Gini(S) = 1 - [(\frac{5}{5})^2] = 0$
- For **Parents = No,** there are **2 examples with "Tennis", 1 example with "Stay in", "Shopping" and "Cinema" each**
- $Gini(S) = 1 - [(\frac{2}{5})^2 + (\frac{1}{5})^2 + (\frac{1}{5})^2 + (\frac{1}{5})^2] =$ 0.72
- $\boldsymbol{Weighted\ Average(Parents)}$

$$= \mathbf{0 * \left(\frac{5}{10}\right) + [0.72 * \left(\frac{5}{10}\right) = 0.36}$$

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Computation of Gini Index for Weather Attribute
- It has three possible values of Sunny (3 examples), Rainy (3 examples) and Windy (4 examples).
- For **Weather = Sunny**, there are **2 examples with "Cinema"** and **1 with "Tennis".**
- $Gini(Sunny) = 1 - [(\frac{2}{3})^2 + (\frac{1}{3})^2] = 0.444$
- For **Weather = Rainy**, there are **2 examples with "Cinema"** and **1 example with "Stay in"**
- $Gini(Rainy) = 1 - [(\frac{2}{3})^2 + (\frac{1}{3})^2] = 0.444$
- For **Weather = Windy**, there are **3 examples with "Cinema"** and **1 example with "Shopping"**
- $Gini(Windy) = 1 - [(\frac{3}{4})^2 + (\frac{1}{4})^2] = 0.375$

Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay In
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

$$Weighted\ Average(Weather)$$

$$= 0.444 * \left(\frac{3}{10}\right) + 0.444 * \left(\frac{3}{10}\right) + 0.375 * \left(\frac{4}{10}\right)$$

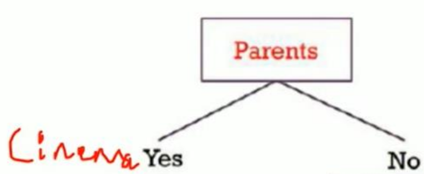
$$= 0.416$$

For Weather - Gini Index: 0.416

For Parents - Gini Index: 0.36 ✓

For Money - Gini Index: 0.486

Parents is selected as it has smallest Gini index.



Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W6	Rainy	Yes	Poor	Cinema
W9	Windy	Yes	Rich	Cinema

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No |Weather Attribute

- **Sunny (2 examples)**
- For Parent= No |Weather = Sunny, there are 2 example with "Tennis.
- $Gini(S) = 1 - [\left(\frac{2}{2}\right)^2] = 0$

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No |Weather Attribute

- **Rainy (1 example).**
- For Parents = No | Weather = Rainy, there is 1 example with "Stay In".
- $Gini(S) = 1 - [\left(\frac{1}{1}\right)^2] = 0$

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No |Weather Attribute

- **Windy (2 example)**
- For Parents = No | Weather = Windy, there is 1 example with "Cinema" and 1 example with "Shopping".
- $Gini(S) = 1 - \left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right] = 0.5$

$$Weighted\ Average(Parents = No\,|\,Weather) = 0 * \left(\frac{2}{5}\right) + 0 * \left(\frac{1}{5}\right) + 0.5 * \left(\frac{2}{5}\right) = 0.2$$

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No|Money Attribute

- Rich (4 examples)
- For Parents = No| Money = Rich, there is 1 example with "stay in" and "Shopping" each and 2 examples of "Tennis".
- $Gini(S) = 1 - [\left(\frac{1}{4}\right)^2 + \left(\frac{1}{4}\right)^2 + \left(\frac{2}{4}\right)^2] = 0.625$

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents = No|Money Attribute

- Poor (1 example)
- For Parents = No| Money = Poor, there is 1 example with "Cinema".
- $Gini(S) = 1 - [\left(\frac{1}{1}\right)^2] = 0$
- $Weighted\ Average\ (Parents = No|\ Money) = 0.625 * (4/5) + 0 * (1/5) = 0.5$

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

For Parents = No | Weather - Gini Index: 0.2

For Parents = No | Money - Gini Index: 0.5

Weather is selected as it has smallest Gini index.

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

Now, for Parent=No & Weather=Sunny, we have all instances as Tennis.

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W10	Sunny	No	Rich	Tennis

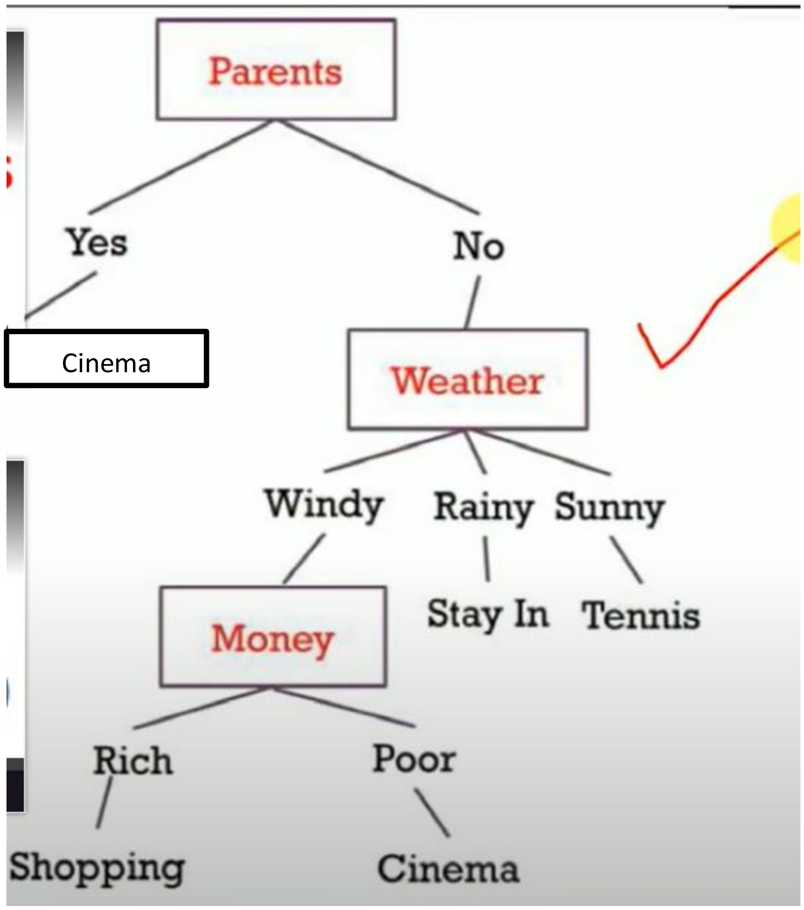
Now, for Parents=No & Weather=Rainy, we have all instances as Stay In.

Weekend	Weather	Parents	Money	Decision
W5	Rainy	No	Rich	Stay In

Now, for Parent=No & Weather=Windy, we need to split.

Weekend	Weather	Parents	Money	Decision
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping

Decision Tree for the given data set.


Parents
Yes
No
Cinema
Weather
Windy
Rainy
Sunny
Money
Stay In
Tennis
Rich
Poor
Shopping
Cinema

Linear Regression Numerical:

The least square regression line for the set of n data points is given by the equation of a line in slope intercept form:

$$y = a x + b$$

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$b = \frac{1}{n}(\sum_{i=1}^{n} y_i - a\sum_{i=1}^{n} x_i)$$

Q.1

The values of y and their corresponding values of y are shown in the table below

x	0	1	2	3	4
y	2	3	5	4	6

a) Find the least square regression line $y = a x + b$.

b) Estimate the value of y when $x = 10$.

Solution:

We use a table to calculate a and b.

x	**y**	**x y**	**x^2**
0	2	0	0
1	3	3	1
2	5	10	4
3	4	12	9
4	6	24	16
$\Sigma x = 10$	$\Sigma y = 20$	$\Sigma x\, y = 49$	$\Sigma x^2 = 30$

We now calculate a and b using the least square regression formulas for a and b.

$a = (n\Sigma x\, y - \Sigma x \Sigma y) / (n\Sigma x^2 - (\Sigma x)^2) = (5*49 - 10*20) / (5*30 - 10^2) = 0.9$

$b = (1/n)(\Sigma y - a\ \Sigma x) = (1/5)(20 - 0.9*10) = 2.2$

b) Now that we have the least square regression line $y = 0.9\ x + 2.2$, substitute x by 10 to find the value of the corresponding y.

$y = 0.9 * 10 + 2.2 = 11.2$

Q.2

a) Find the least square regression line for the following set of data

$$\{(-1\ ,\ 0),(0\ ,\ 2),(1\ ,\ 4),(2\ ,\ 5)\}$$

b) Plot the given points and the regression line in the same rectangular system of axes.

Solution:

We use a table as follows

x	**y**	**x y**	**x^2**
-1	0	0	1
0	2	0	0
1	4	4	1
2	5	10	4
$\Sigma x = 2$	$\Sigma y = 11$	$\Sigma x\ y = 14$	$\Sigma x^2 = 6$

We now use the above formula to calculate a and b as follows

$a = (n\Sigma x\, y - \Sigma x \Sigma y) / (n\Sigma x^2 - (\Sigma x)^2) = (4*14 - 2*11) / (4*6 - 2^2) = 17/10 = 1.7$

$b = (1/n)(\Sigma y - a\, \Sigma x) = (1/4)(11 - 1.7*2) = 1.9$

b) We now graph the regression line given by y = ax + b and the given points.

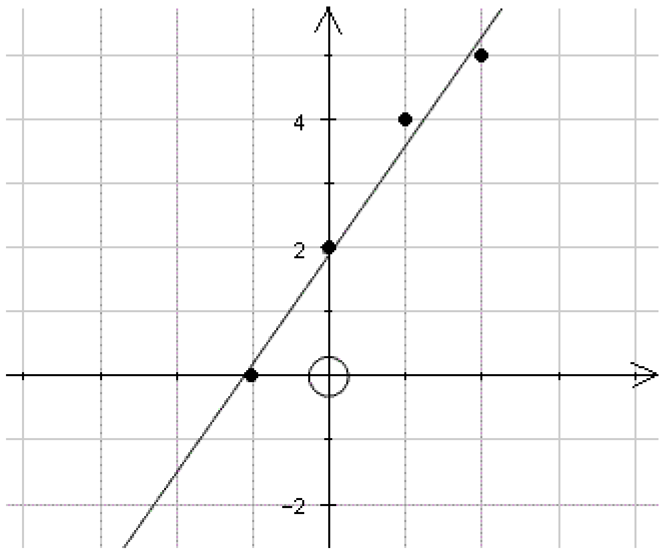


Figure 4. Graph of linear regression in problem 2.

Q.3

Question: Find linear regression equation for the following two sets of data:

x	2	4	6	8
y	3	7	5	10

Solution:

Construct the following table:

x	y	x^2	xy
2	3	4	6
4	7	16	28
6	5	36	30
8	10	64	80
$\sum x$ = 20	$\sum y$ = 25	$\sum x^2$ = 120	$\sum xy$ = 144

b

=

$\frac{n\sum xy-(\sum x)(\sum y)}{n\sum x^2-(\sum x)^2}$

b

=

$\frac{4\times 144-20\times 25}{4\times 120-400}$

b = 0.95

$a=\frac{\sum y\sum x^2-\sum x\sum xy}{n(\sum x^2)-(\sum x)^2}$

$a=\frac{25\times 120-20\times 144}{4(120)-400}$

a = 1.5

Linear regression is given by:

y = a + bx

y = 1.5 + 0.95 x