<table>
<tr><td>ASSIGNMENT NO</td><td>A2</td></tr>
<tr><td>TITLE</td><td>To find the decision based on a given scenario from a dataset using Decision Tree Classifier.</td></tr>
<tr><td>PROBLEM STATEMENT/ DEFINITION</td><td>A dataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future. Find the root node of decision tree. According to the decision tree you have made from previous training data set, what is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?
<table>
<tr><th>ID</th><th>Age</th><th>Income</th><th>Gender</th><th>Marital Status</th><th>Buys</th></tr>
<tr><td>1</td><td>< 21</td><td>High</td><td>Male</td><td>Single</td><td>No</td></tr>
<tr><td>2</td><td>< 21</td><td>High</td><td>Male</td><td>Married</td><td>No</td></tr>
<tr><td>3</td><td>21-35</td><td>High</td><td>Male</td><td>Single</td><td>Yes</td></tr>
<tr><td>4</td><td>>35</td><td>Medium</td><td>Male</td><td>Single</td><td>Yes</td></tr>
<tr><td>5</td><td>>35</td><td>Low</td><td>Female</td><td>Single</td><td>Yes</td></tr>
<tr><td>6</td><td>>35</td><td>Low</td><td>Female</td><td>Married</td><td>No</td></tr>
<tr><td>7</td><td>21-35</td><td>Low</td><td>Female</td><td>Married</td><td>Yes</td></tr>
<tr><td>8</td><td>< 21</td><td>Medium</td><td>Male</td><td>Single</td><td>No</td></tr>
<tr><td>9</td><td><21</td><td>Low</td><td>Female</td><td>Married</td><td>Yes</td></tr>
<tr><td>10</td><td>> 35</td><td>Medium</td><td>Female</td><td>Single</td><td>Yes</td></tr>
<tr><td>11</td><td>< 21</td><td>Medium</td><td>Female</td><td>Married</td><td>Yes</td></tr>
<tr><td>12</td><td>21-35</td><td>Medium</td><td>Male</td><td>Married</td><td>Yes</td></tr>
<tr><td>13</td><td>21-35</td><td>High</td><td>Female</td><td>Single</td><td>Yes</td></tr>
<tr><td>14</td><td>> 35</td><td>Medium</td><td>Male</td><td>Married</td><td>No</td></tr>
</table>
</td></tr>
<tr><td>OBJECTIVE</td><td>To understand how to decision tree classifier algorithm works on the give dataset</td></tr>
<tr><td>OUTCOME</td><td>To find the decision based on a given scenario of people with income, gender and marital status information from a dataset using Decision Tree Classifier.</td></tr>
<tr><td>S/W PACKAGES AND HARDWARE APPARATUS USED</td><td>Core 2 DUO/i3/i5/i7 64-bit processor
OS-LINUX 64 bit OS
Editor-gedit/Eclipse
S/W- Jupyter Notebook/Weka/Python</td></tr>
<tr><td>REFERENCES</td><td>1. Giuseppe Bonaccorso, “ Machine Learning Algorithms”, Packt Publishing Limited, ISBN-10: 1785889621, ISBN-13: 978-1785889622
2. Josh Patterson, Adam Gibson, “Deep Learning : A</td></tr>
</table>

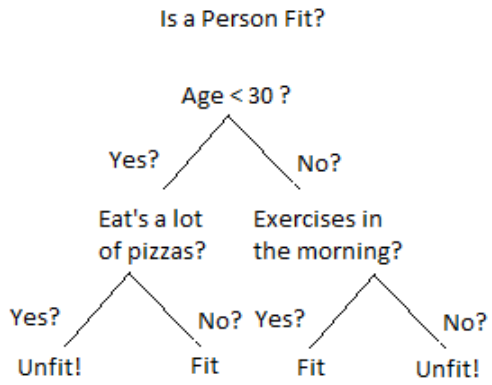
	Practitioners Approach", O'REILLY, SPD, ISBN: 978-93-5213-604-9, 2017 Edition 1st. 3. Nikhil Buduma, "Fundamentals of Deep Learning", O'REILLY publication, Second Edition, 2017,ISBN: 1491925612
STEPS	1. Place the best attribute of the dataset at the root of the **tree**. 2. Split the training set into subsets. 3. Repeat **step** 1 and **step** 2 on each subset until you find leaf nodes in all the branches of the **tree**.
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment No. 3. Problem Definition 4. Learning Objective 5. Learning Outcome 6. Concepts Related Theory 7. Algorithm 8. Test Cases 9. Conclusion/Analysis

- **Prerequisites:** Basic knowledge about Algorithms and any programming knowledge Java/python
- **Concepts related Theory**

A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

Decision Tree consists of :

1. **Nodes** : Test for the value of a certain attribute.
2. **Edges/ Branch** : Correspond to the outcome of a test and connect to the next node or leaf.
3. **Leaf nodes** : Terminal nodes that predict the outcome (represent class labels or class distribution).



To understand the concept of Decision Tree consider the above example. Let's say you want to predict whether a person is fit or unfit, given their information like age, eating habits, physical activity, etc. The decision nodes are the questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves represent outcomes like either 'fit', or 'unfit'.

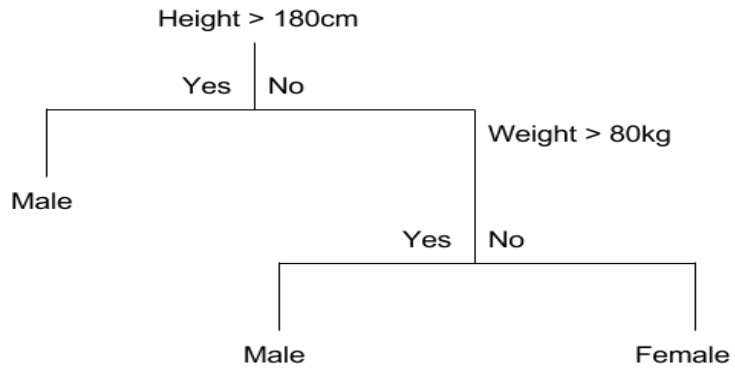
There are two main types of Decision Trees:

1. Classification Trees.
2. Regression Trees.

1. Classification trees (Yes/No types) :

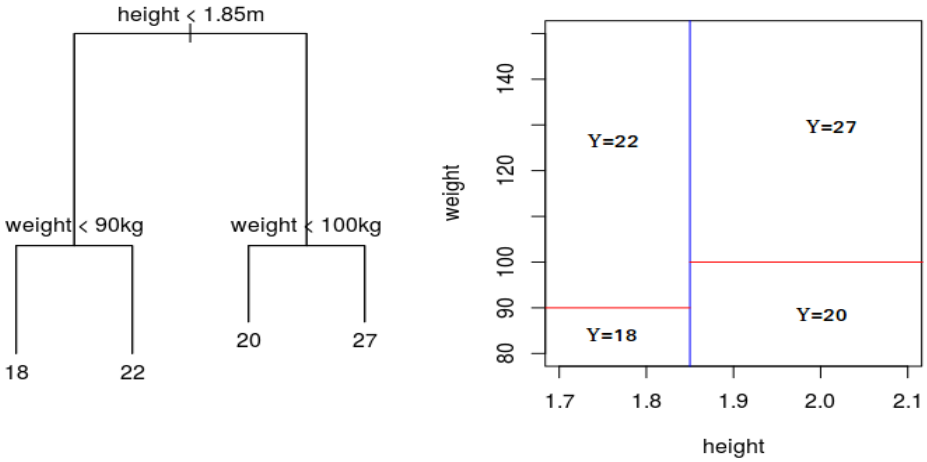
What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is **Categorical/ discrete**.

Such a tree is built through a process known as **binary recursive partitioning**. This is an iterative process of **splitting the data into partitions**, and then splitting it up further on each of the branches.



2. Regression trees (Continuous data types) :

Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. (e.g. the price of a house, or a patient's length of stay in a hospital)



Creation of Decision Tree :

In this method a set of training examples is broken down into smaller and smaller subsets while at the same time an associated decision tree get incrementally developed. At the end of the learning process, a decision tree covering the training set is returned.

The key idea is to use a decision tree to partition the data space into cluster (or dense) regions and empty (or sparse) regions.

In Decision Tree Classification a new example is classified by submitting it to a series of tests that determine the class label of the example. These tests are organized in a hierarchical structure called a decision tree. Decision Trees follow Divide-and-Conquer Algorithm.

Divide and Conquer

Decision trees are built using a heuristic called **recursive partitioning**. This approach is also commonly known as **divide and conquer** because it splits the data into subsets, which are then split repeatedly into even **smaller subsets**, and so on and so forth until the process stops when the algorithm determines the data within the subsets are **sufficiently homogenous**, or another stopping criterion has been met.

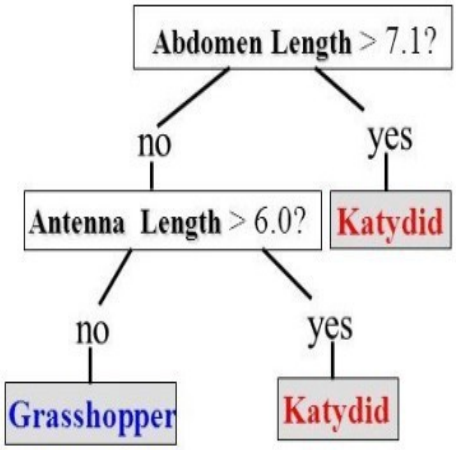
Basic Divide-and-Conquer Algorithm :

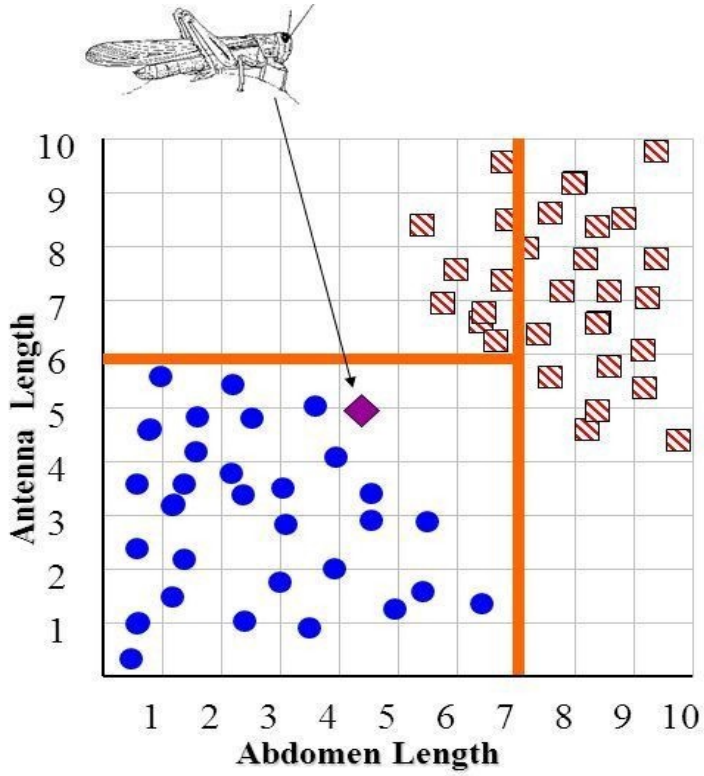
1. Select a test for root node. Create branch for each possible outcome of the test.
2. Split instances into subsets. One for each branch extending from the node.
3. Repeat recursively for each branch, using only instances that reach the branch.
4. Stop recursion for a branch if all its instances have the same class.

Decision Tree Classifier

- Using the decision algorithm, we start at the tree root and split the data on the feature that results in the **largest information gain (IG)** (reduction in uncertainty towards the final decision).

- In an iterative process, we can then repeat this splitting procedure at each child node **until the leaves are pure**. This means that the samples at each leaf node all belong to the same class.
- In practice, we may set a **limit on the depth of the tree to prevent overfitting**. We compromise on purity here somewhat as the final leaves may still have some impurity.





Attribute Selection Measures

Attribute selection measure is a heuristic for selecting the splitting criterion that partition data into the best possible manner. It is also known as splitting rules because it helps us to determine breakpoints for tuples on a given node. ASM provides a rank to each feature(or attribute) by explaining the given dataset. Best score attribute will be selected as a splitting attribute. In the case of a continuous-valued attribute, split points for branches also need to define. Most popular selection measures are Information Gain, Gain Ratio, and Gini Index.

Information Gain

Shannon invented the concept of entropy, which measures the impurity of the input set. In physics and mathematics, entropy referred as the randomness or the impurity in the system. In information theory, it refers to the impurity in a group of examples. Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. ID3 (Iterative Dichotomiser) decision tree algorithm uses information gain.

$$\text{Info}(D) = -\sum_{i=1}^{m} pi \log_2 pi$$

Where, Pi is the probability that an arbitrary tuple in D belongs to class Ci.

$$\text{Info}_A(D) = \sum_{j=1}^{V} \frac{|Dj|}{|D|} \times \text{Info}(D_j)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

Where,

- Info(D) is the average amount of information needed to identify the class label of a tuple in D.
- |Dj|/|D| acts as the weight of the jth partition.

- InfoA(D) is the expected informa-tion required to classify a tuple from D based on the partitioning by A.

The attribute A with the highest information gain, Gain(A), is chosen as the splitting attribute at node N().

Gain Ratio

Information gain is biased for the attribute with many outcomes. It means it prefers the attribute with a large number of distinct values. For instance, consider an attribute with a unique identifier such as customer_ID has zero info(D) because of pure partition. This maximizes the information gain and creates useless partitioning.

C4.5, an improvement of ID3, uses an extension to information gain known as the gain ratio. Gain ratio handles the issue of bias by normalizing the information gain using Split Info. Java implementation of the C4.5 algorithm is known as J48, which is available in WEKA data mining tool.

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

Where,

- |Dj|/|D| acts as the weight of the jth partition.
- v is the number of discrete values in attribute A.

The gain ratio can be defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

The attribute with the highest gain ratio is chosen as the splitting attribute (Source).

Gini index

Another decision tree algorithm CART (Classification and Regression Tree) uses the Gini method to create split points.

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

Where, pi is the probability that a tuple in D belongs to class Ci.

The Gini Index considers a binary split for each attribute. You can compute a weighted sum of the impurity of each partition.If a binary split on attribute A partitions data D into D1 and D2, the Gini index of D is:

$$\text{Gini}_A(D)=\frac{|D1|}{|D|}\text{Gini}(D_1)+\frac{|D2|}{|D|}\text{Gini}(D_2)$$

In case of a discrete-valued attribute, the subset that gives the minimum gini index for that

chosen is selected as a splitting attribute. In the case of continuous-valued attributes, the strategy is to select each pair of adjacent values as a possible split-point and point with smaller gini index chosen as the splitting point.

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

The attribute with minimum Gini index is chosen as the splitting attribute.

Advantages of Classification with Decision Trees:

1. Inexpensive to construct.
2. Extremely fast at classifying unknown records.
3. Easy to interpret for small-sized trees
4. Accuracy comparable to other classification techniques for many simple data sets.
5. Excludes unimportant features.

Disadvantages of Classification with Decision Trees:

1. Easy to overfit.
2. Decision Boundary restricted to being parallel to attribute axes.

3. Decision tree models are often biased toward splits on features having a large number of levels.
4. Small changes in the training data can result in large changes to decision logic.
5. Large trees can be difficult to interpret and the decisions they make may seem counter intuitive.

Applications of Decision trees in real life :

1. Biomedical Engineering (decision trees for identifying features to be used in implantable devices).
2. Financial analysis (Customer Satisfaction with a product or service).
3. Astronomy (classify galaxies).
4. System Control.
5. Manufacturing and Production (Quality control, Semiconductor manufacturing, etc).
6. Medicines (diagnosis, cardiology, psychiatry).
7. Physics (Particle detection).

Test Cases:

Split the dataset into training and testing dataset