Decision trees

Definition:

**Decision tree learning** is a supervised learning approach used in statistics, data mining and machine learning. In this formalism, a classification or regression decision tree is used as a predictive model to draw conclusions about a set of observations.

Concept:

Tree models where the target variable can take a discrete set of values are called **classification trees** Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**

A decision tree is a **non-parametric** supervised learning algorithm which means it does not make such assumptions about the functional form of the relationship. Instead, they tend to be more flexible and can adapt to the complexity of the data.

## **Types of Decision Trees**

There are **multiple algorithms** (with extremely long, and scary names) which are used for building Decision Trees. Some of them are :

* [**ID3**](https://en.wikipedia.org/wiki/ID3_algorithm)(Iterative Dichotomiser 3)
* [**C4.5**](https://en.wikipedia.org/wiki/C4.5_algorithm)(successor of ID3)
* [**CART**](https://en.wikipedia.org/wiki/Predictive_analytics#Classification_and_regression_trees_.28CART.29) (Classification And Regression Tree)
* [**Chi-square automatic interaction detection**](https://en.wikipedia.org/wiki/Chi-square_automatic_interaction_detection) (CHAID). Performs multi-level splits when computing classification trees.
* [**MARS**](https://en.wikipedia.org/wiki/Multivariate_adaptive_regression_splines): extends decision trees to handle numerical data better.

Many of these improvements of basic decision trees were incorporated to other tree-based methods such as random forests and gradient boosted trees

### some of the terminologies.

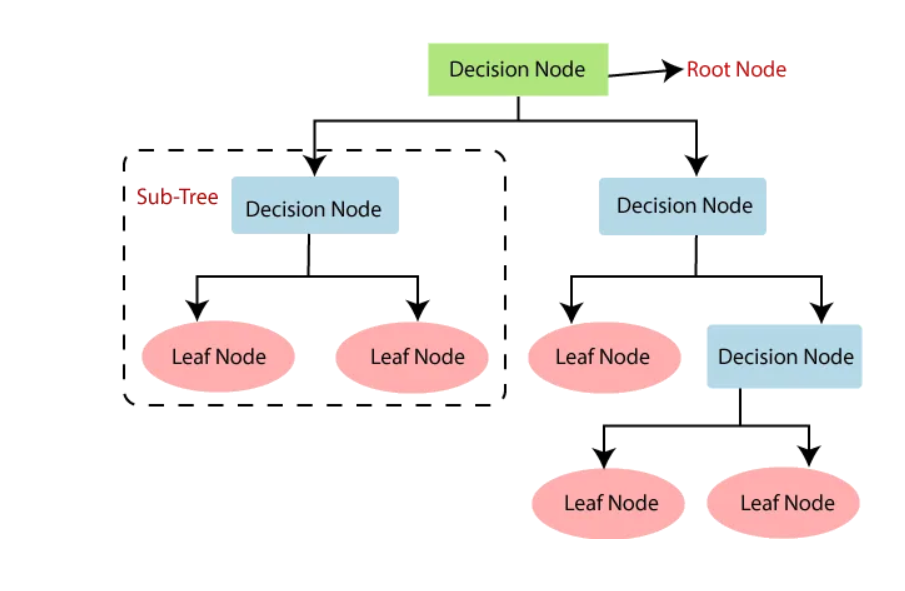
Root Nodes — It is the node present at the beginning of a decision tree from this node the population starts dividing according to various features.

Decision Nodes — the nodes we get after splitting the root nodes are called Decision Node

Leaf Nodes — the nodes where further splitting is not possible are called leaf nodes or terminal nodes

Branch/Sub-tree — just like a small portion of a graph is called sub-graph similarly a sub-section of this decision tree is called sub-tree.

Pruning — is nothing but cutting down some nodes to stop overfitting.



**IDEA** was to represent data as a tree where each internal node denotes a test on an attribute (basically a condition), each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Several criteria can be used to evaluate splits in decision trees. The most common ones include:

1. **Gini Impurity**: Gini impurity measures the probability of incorrectly classifying a randomly chosen element if it were randomly labeled according to the distribution of labels in the node. It is calculated as the sum of the squared probabilities of each class being chosen, with a lower value indicating a purer node
2. **Entropy**: Entropy measures the impurity or randomness of a dataset's distribution. In the context of decision trees, it quantifies the uncertainty of a node's class distribution. A lower entropy value implies a more homogeneous distribution of classes in the node.
3. **Information Gain**: Information gain is the difference between the entropy or impurity of the parent node and the weighted sum of the impurities of the child nodes after the split. It represents the reduction in entropy achieved by splitting the dataset on a particular feature. Higher information gain indicates a better split.

## **After looking at the example above, the following questions might arise :**

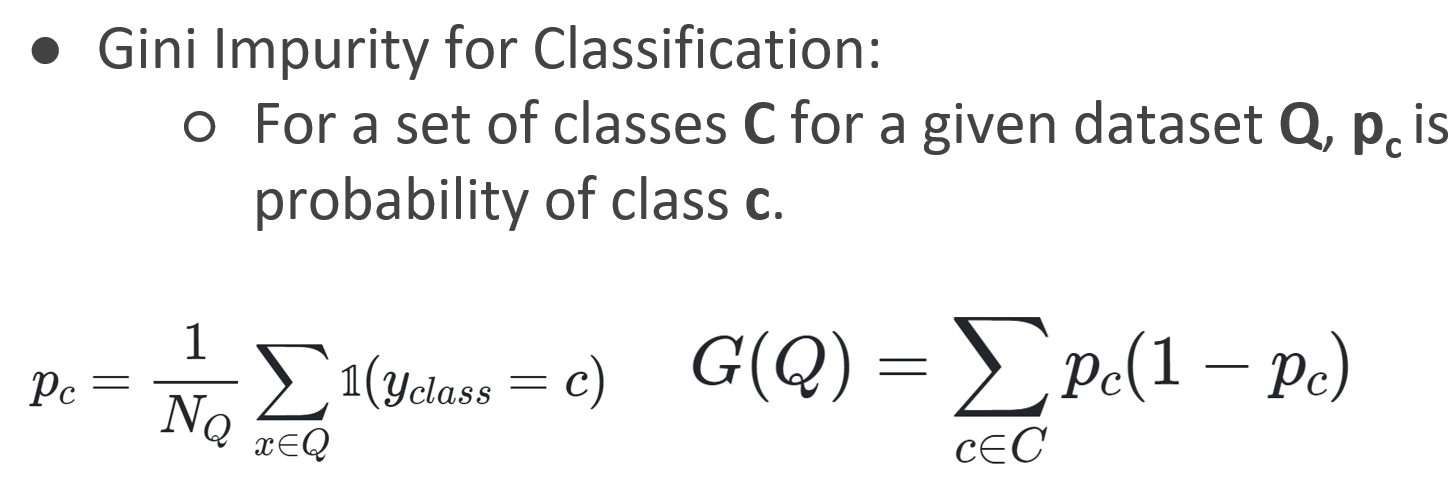
1. **Which feature should we choose first?**
   * The feature selection process starts at the root node. The goal is to choose the feature that best separates the data into different classes or categories. This is typically done by evaluating various splitting criteria, such as Gini impurity or information gain, for each feature.
2. **What should be the order of splitting?**
   * The order of splitting depends on the chosen splitting criterion. At each node, the algorithm evaluates all possible splits on all features and chooses the split that maximizes the chosen criterion. This process is repeated recursively for each child node until a stopping criterion is met (e.g., reaching a maximum depth, minimum number of samples per leaf).
3. **How do we decide the number of child nodes?**
   * The number of child nodes is determined by the number of unique values or categories in the feature being split. For example, if a feature has three unique values after splitting, there will be three child nodes corresponding to each value.
4. **How can we compare two features w.r.t which one is better for a split?**
   * Features are compared based on their ability to reduce impurity or increase information gain at each node. The feature that results in the highest impurity reduction or information gain is chosen for the split. Different splitting criteria, such as Gini impurity, entropy, or misclassification error, can be used to evaluate the quality of splits.
5. **How do we decide the splitting criteria?**
   * The splitting criteria are chosen based on the specific objectives of the problem and the characteristics of the dataset. Common splitting criteria include:
     + Gini impurity: Measures the probability of incorrectly classifying a randomly chosen element if it were randomly labeled according to the distribution of labels in the node.
     + Entropy: Measures the randomness or uncertainty of a dataset's distribution.
     + Information gain: Measures the reduction in entropy or impurity achieved by splitting the dataset on a particular feature.

That was a brief about decision trees, now let’s focus on our project:

**Main Idea:**

We used classification decision tree (CART) with criterion of splitting is (Gini Impurity or Entropy)

1)

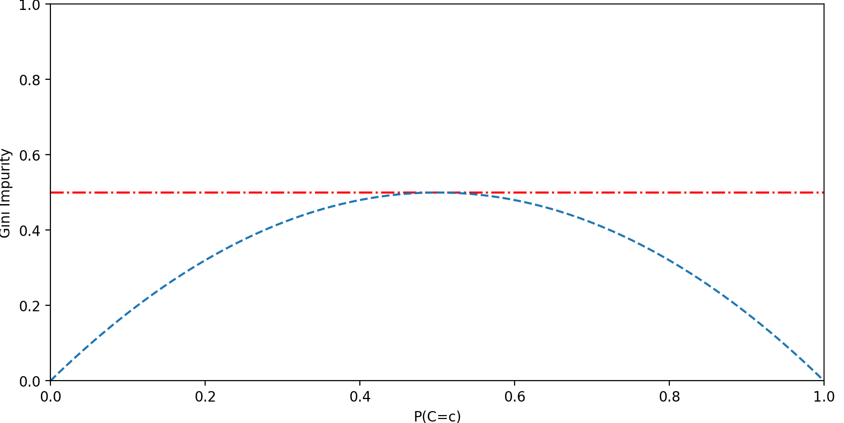


●If the goal of a decision tree is to separate out classes, we can use **Gini impurity** to decide on data split values.

●We want to **minimize** the Gini impurity at leaf nodes.

●Minimized impurity at leaf nodes means we are separating classes effectively!

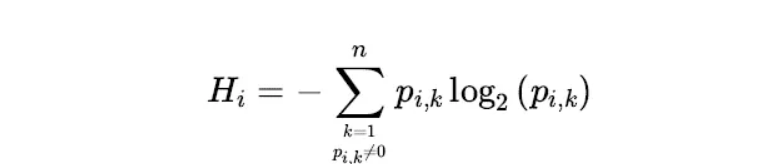
**NOTE**: The maximum value for Gini-impurity is ½;



***2)***

***Entropy:*** Entropy represents the order of randomness. In decision tree, it helps model in selection of feature for splitting, at the node by measuring the purity of the split. If,

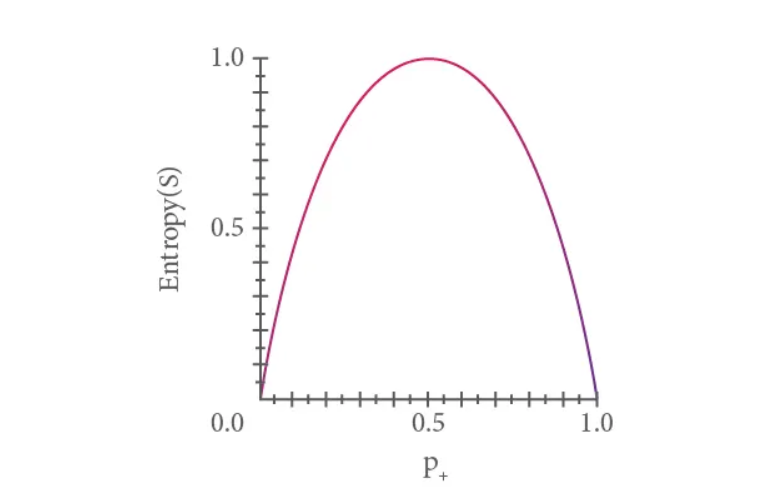
* Entropy = 0 means it is pure split i.e., all instances are of only 1 class.
* Entropy=1 means Completely impure split i.e., equal instances (50%–50%) of both class at node causing extreme disorder.



Where, p(i,k ) is the probability of positive and negative class i at particular node.

n=Number of distinct class value at specific node.

The range of entropy H varies between 0–1.



## **About Dataset**

**Data Description:**  
The file Bank.xls contains data on 5000 customers. The data include customer demographic information (age, income, etc.), the customer's relationship with the bank (mortgage, securities account, etc.), and the customer response to the last personal loan campaign (Personal Loan). Among these 5000 customers, only 480 (= 9.6%) accepted the personal loan that was offered to them in the earlier campaign.

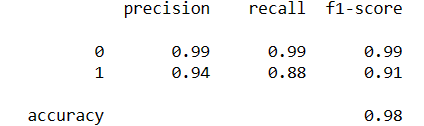
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  | **Data Description:** | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | ID | Customer ID | |  |  |  |  |  |  |  |
|  | Age | Customer's age in completed years | | | |  |  |  |  |  |
|  | Experience | #years of professional experience | | | |  |  |  |  |  |
|  | Income | Annual income of the customer ($000) | | | | |  |  |  |  |
|  | ZIPCode | Home Address ZIP code. | | |  |  |  |  |  |  |
|  | Family | Family size of the customer | | | |  |  |  |  |  |
|  | CCAvg | Avg. spending on credit cards per month ($000) | | | | | |  |  |  |
|  | Education | Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional | | | | | | | |  |
|  | Mortgage | Value of house mortgage if any. ($000) | | | | |  |  |  |  |
|  | Personal Loan | Did this customer accept the personal loan offered in the last campaign? | | | | | | | |  |
|  | Securities Account | Does the customer have a securities account with the bank? | | | | | | |  |  |
|  | CD Account | Does the customer have a certificate of deposit (CD) account with the bank? | | | | | | | |  |
|  | Online | Does the customer use internet banking facilities? | | | | | |  |  |  |
|  | CreditCard | Does the customer use a credit card issued by UniversalBank? | | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

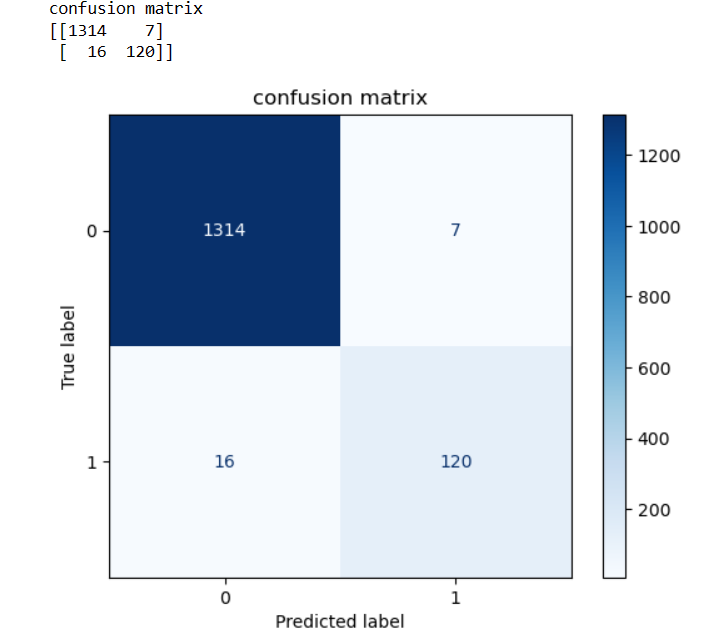
Implementation: In notebook

Cross validation was used with 10 folds:

1. **Train-Test Split**:
   * Training set: 70% of the original data.
   * Test set: 30% of the original data.
2. **Cross-Validation**:
   * Each fold in 10-fold cross-validation will use 90% of the training data for training and 10% for validation.
   * So, for each fold:
     + Training set: 90% of 70% of the original data = 63% of the original data.
     + Validation set: 10% of 70% of the original data = 7% of the original data.

Results:





That was the result after using grid search and that choose the following parameters:

criterion: Entropy

# The function to measure the quality of a split

**Hyperparameters used:**

max\_depth: 4

# The maximum depth of the tree

min\_samples\_leaf:3

# The minimum number of samples required to be at a leaf node

min\_samples\_split: 2

# The minimum number of samples required to split an internal node