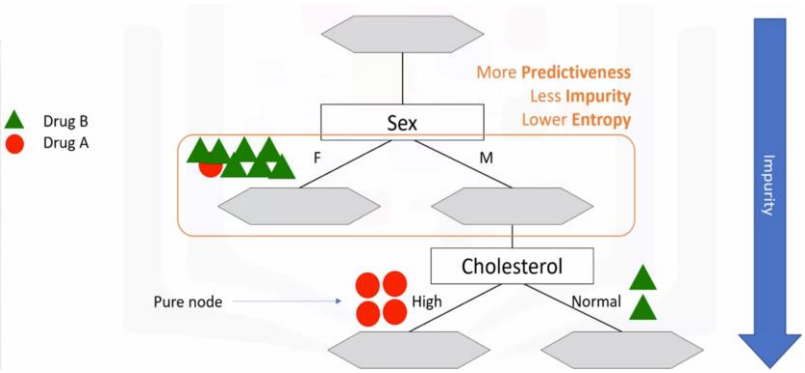
DECISION TREES

Decision Trees can be described as

- Decision Trees are built by splitting the training set into distinct nodes where one node contains all of or most of, one category of the data
- Decision Trees are about testing an attribute and branching the cases based on the result of the test
- Each **internal node** corresponds to a test and each branch corresponds to a result of the test and **each leaf** node assigns an observation to a class
- Here is the way that a Decision Tree is built
    - A Decision Tree can be constructed by considering the attributes one by one
    - Choose an attribute from our dataset
    - Calculate the significance of the attribute in the splitting if the data
    - Split the data based on the value of the best attribute
    - Then go to each branch and repeat it for the rest of the attributes
    - After building this tree, we can use it to predict the class of unknown cases

Building Decision Trees

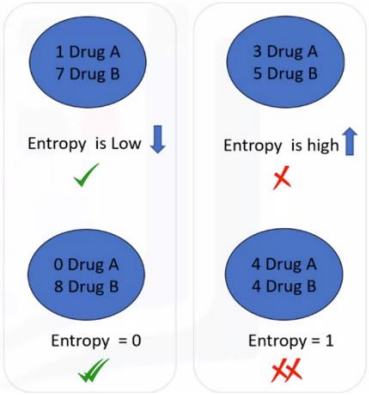
Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A
p15	Middle-age	F	Low	Normal	?



As we can see the example of dataset above, we will build a describe each step of Decision Tree base on that data set

- Decision Trees are built using recursive partitioning to classify the data
- Let say's we have 15 observations in our dataset, the algorithm chooses the most predictive feature to split the data
    - The important thing in making a Decision Tree is to determine which attribute is the best or more predictive to split data based on the feature
    - For example, we pick "Cholesterol" as the first attribute to split data then it will split the data into 2 branches ('High' and 'Normal') since this attribute only have 2 categorical values
        - If the observation has high cholesterol, we can't say with high confidence that Drug B might be suitable for it
        - Also, if the observation's cholesterol is normal, we still don't have sufficient evidence or information to determine if either drug A or Drug B is in fact suitable
        - This is a sample of bad attributes selection for splitting data so try another attribute
    - We repeat the process in the above until we get the best attribute for the parent and find another attribute to split our data again in the branch as the child node
    - The choice of attribute to split data is very important and it is all about purity of the leaves after the split
    - A node in the tree is considered pure if in 100% of the cases, the nodes fall into a specific category of the target field
    - In fact, the method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step

- The best Decision Trees should
  - More **Predictiveness**
    - Predictiveness is based on decrease in impurity of nodes
  - Less **Impurity**
    - We're looking for the best feature to decrease the impurity of the observations in the leaves after splitting them up based on that feature
    - Impurity of nodes is calculated by entropy of data in the node
  - Lower **Entropy**
    - Entropy is the amount of information disorder or the amount of randomness in the data
    - The entropy in the node depends on how much random data is in that node and is calculated for each node
    - In Decision Trees we're looking for the trees that have the **smallest entropy** in their nodes
    - The entropy is used to calculate the homogeneity of the samples in that node
      - If the samples are completely homogenous, the entropy is 0 and if the samples are equally divided it has an entropy of 1



    - We can easily calculate the entropy of a node using the frequency table of the attribute through the entropy formula where:
      - $Entropy = -p(A)2\log(p(A)) - p(B)2\log(p(B))$
      - $\boldsymbol{p}$ is for the proportion or ratio of a category (such as drug A or drug B)
    - This is easily calculated by the libraries or packages that we will use
  - Higher **Information Gain**
    - Information Gain is the information that can increase the level of certainty after splitting
    - This equation is $IG = (Entropy\ before\ split) - (weighted\ entropy\ after\ split)$
    - The Information Gain value is calculated using Entropy values of Parent and child nodes
    - Once we get all the Entropy values for each attribute, we can use Information Gain to choose the best parameter based on the **Highest Information Gain after splitting**
    - We can think of Information Gain and Entropy as opposites means as **Entropy value decreases then the Information Gain increases and vice versa**
    - So, constructing a Decision Tree is all about finding attributes that return the highest Information Gain

Here is the example of how to choose the best parameter based on **Entropy** and **Information Gain** values

(note in excel: Entropy=-9/14*(LOG(9/14,2))-5/14*(LOG(5/14,2)))