1. Decision Tree Classifier
2. A decision is tree is a tree where each node represents the features, the branches represent a decision rule, and the leaf nodes represent the outcome. Each decision tree has a root which is the starting point of the tree. A decision tree classifier is a two-steps process: Learning step which consist of training the model based on training data set, and prediction step which will consist of testing the model with test data. A decision tree classifier is an example of classification where the target or outcome will be a categorical variable. In other words, decision tree classifier classifies the features in group or class. A decision tree can handle high dimensional data with good accuracy. Decision tree algorithm divides the tree in recursive way. (Datacamp.com)

To perform any decision tree algorithm, we must follow three steps (Datacamp.com):

* Select the best attribute using Attribute Selection Measures (ASM)
* Take that attribute as a decision tree node and divides rest of the dataset into smaller subsets.
* Starts building the tree by repeating recursively this process until there are no more attribute, or all the tuples belong to the same attribute value.

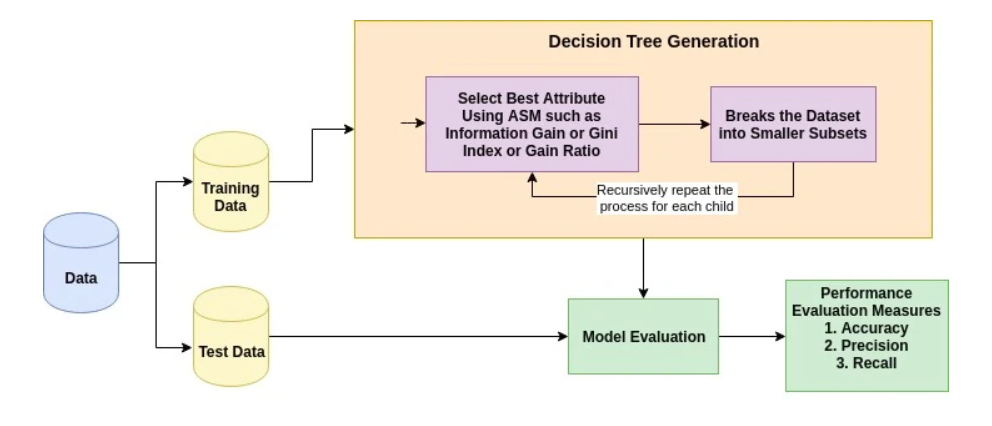


Figure 1: Decision Tree Generation

1. Attribute Selection Measures (ASM)**:** Attribute selection measure is a heuristic for selecting the

and splitting the data set into the best possible to obtain the best attribute. ASM provides a rank to each feature of a given dataset. Therefore, the best score attribute will be selected as a splitting attribute. There are many ways of implementing the Attribute Selection Measures:

* Information gain**:** It computes the difference between entropy (measure the impurity of the set) before split and average entropy after split of the dataset given certain attributes.
* Gain Ratio: It takes the attribute that has large distinct value.
* Gini index: It takes a binary split for each attribute. it will consider the Gini index that is minimum as a splitting attribute

4.a) Decision Tree classifier works both with numerical and categorical with less than 100 categories

4.b) Decision tree classifier is not well suit for small data set. It will cause overfitting problem

4.c) Overfitting problem. Not easy to determine how deep to grow the tree.

5.a) A decision tree helps determine nonlinear pattern and it is very easy to implement because, there is no need to normalize the columns.

5.b)Decision tree is very sensitive to noisy data. We must balance the dataset before creating the decision three to avoid any biased.

6.) Implementation

To run a decision tree classifier, we must import the libraries below.

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.tree import export\_graphviz

from sklearn.externals.six import StringIO

from IPython.display import Image

import pydotplus

We also need to split the data set into two sets: A training set, and test set

And then create a decision tree classifier object

decisionTree =DecisionTreeClassifier()

We train the model by calling the method fit on the object by given the appropriates arguments

We predict the response for the test data by calling the method predict.

We can also measure the accuracy of the decision tree by calling the methods accuracy score by comparing the predicted values and the tested ones.

7.) Evaluation

To determine how well a test condition performs, we need to compare the degree of impurity of the parent before splitting with degree of the impurity of the child nodes after splitting. The larger their difference, the better the test condition.

<https://www.datacamp.com/community/tutorials/decision-tree-classification-python>