Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

Skill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

Machine Learning

Decision Trees

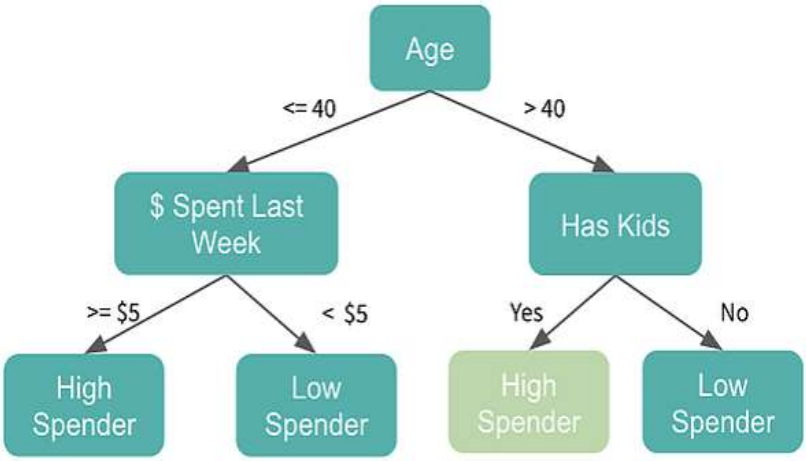
CHAPTER - 8
Decision Trees

Understanding Tree Model

Tree-based machine learning methods are among the most commonly used supervised learning methods. They are constructed by two entities; branches and nodes. Tree-based ML methods are built by recursively splitting a training sample, using different features from a dataset at each node that splits the data most effectively. The splitting is based on learning simple decision rules inferred from the training data.

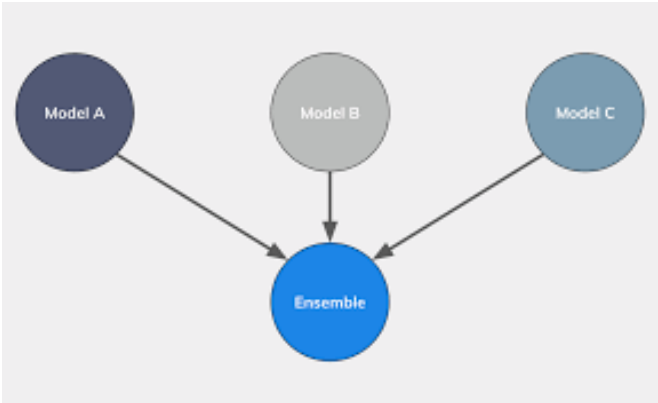
Generally, tree-based ML methods are simple and intuitive; to predict a class label or value, we start from the top of the tree or the root and, using branches, go to the nodes by comparing features on the basis of which will provide the best split.

1.Decision Trees, which are the foundation of all tree-based models.



2.Ensemble Methodsn which include Random Forests,Bagging,Pasting and Boosting methods

as they build many decision trees sequentially or in parallel.



As Tree Modelling is a type of Supervised Learning algorithm having a predefined target variable as it can handle both **continuous or categorical target variables**.Thus this technique splits the data into various homogeneous sets based on most significant splitter or differentiator in input variables

So Lets understand in detail about Tree Terminology:

A tree is a collection of entities called nodes.

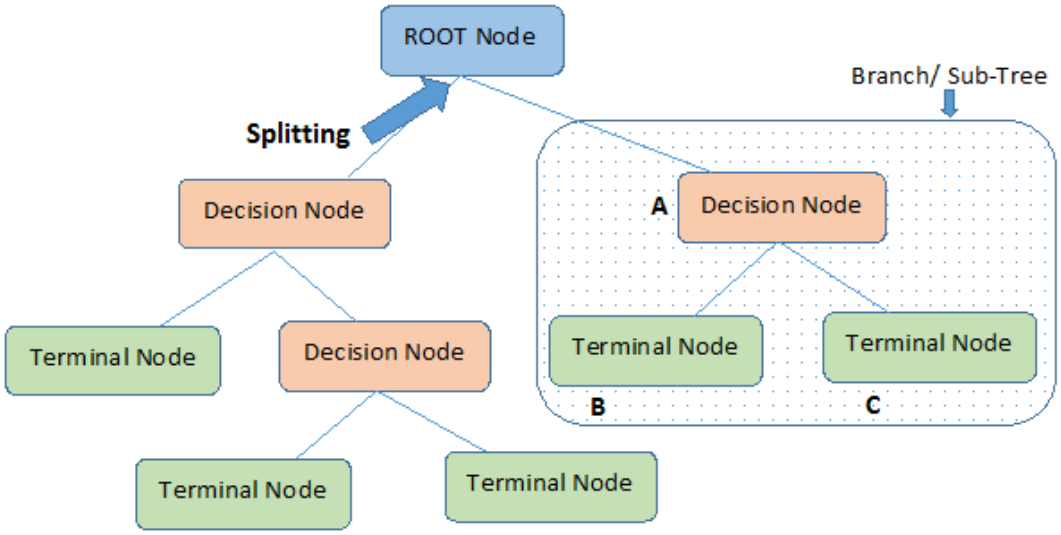
The edges in trees connect the nodes.

The Root is the primary node of the tree.

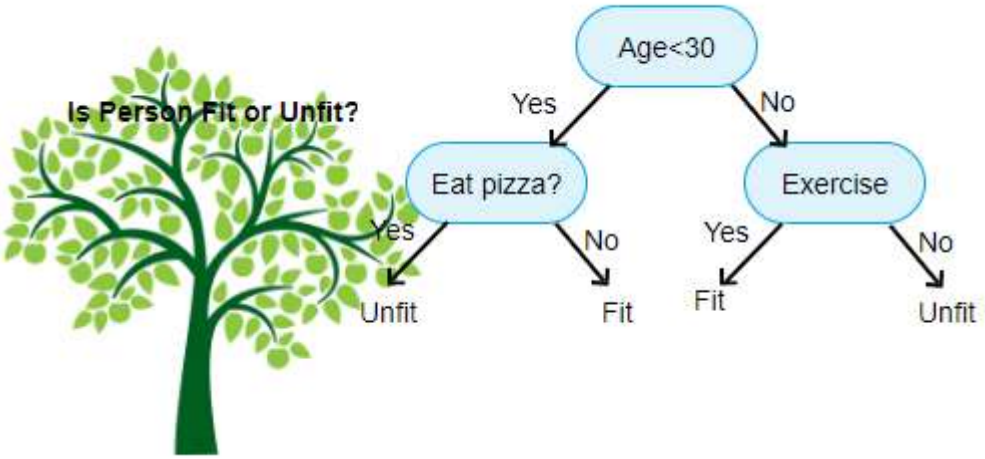
Every node will have its own value or data,and it might or might not possess a child node.

If a root node is connected to any other node,that root will become the Parent node and the connected one will be the child node.

The Terminal nodes or leaves are the last nodes of a tree.



Terminal nodes are the nodes without children. Like real trees,we have root,branches and leaves



Thus Tree based algorithms are considered to be one of the **best and mostly used supervised learning methods.** As they empower predictive models with high accuracy, stability and ease of interpretation.

Understanding Decision Trees:

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

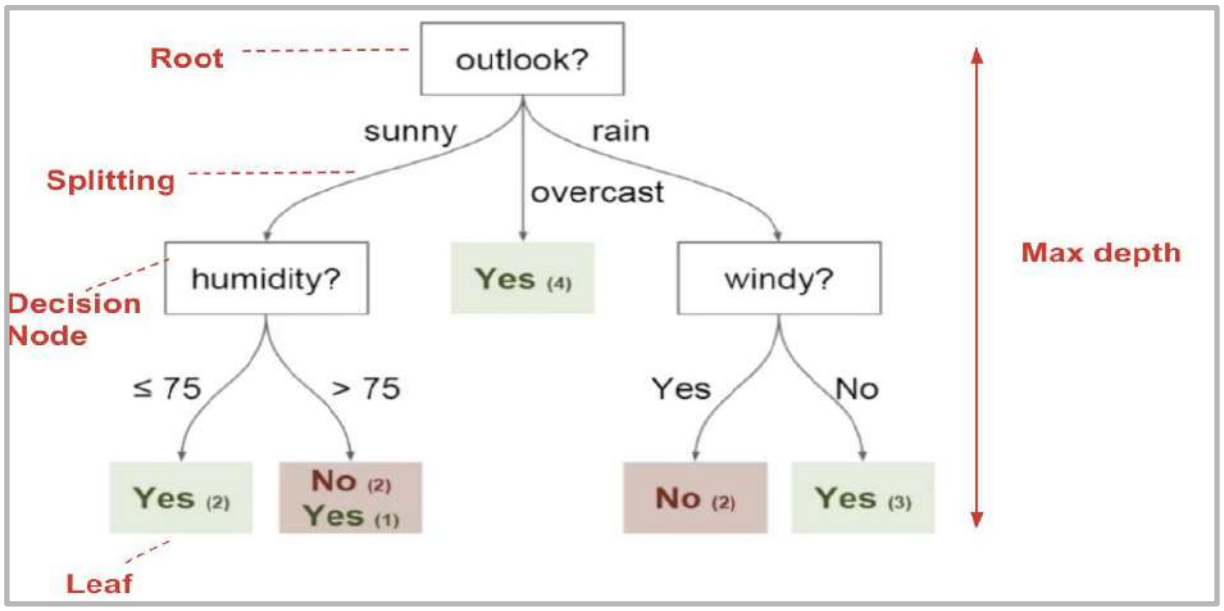
Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric **supervised learning** method used for both **classification** and **regression** tasks.

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**. Classification And Regression Tree (CART) is the general term for this.

So Types of decision trees are based on the type of target variable we have. It can be of two types:

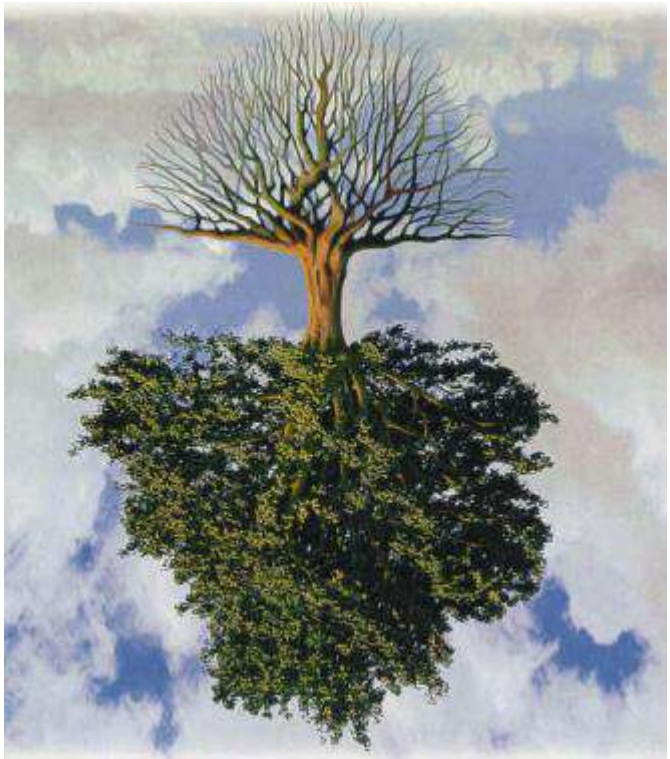
1. **Categorical Variable Decision Tree: where the target variable is categorical** Example:- Suppose we have a scenario of student problem, where the target variable was "Student will play cricket or not" i.e. YES or NO.
2. **Continuous Variable Decision Tree: where the target variable is continuous**

Example:- Let's say we have a problem to predict whether a customer will pay his **renewal premium with an insurance company** where we can build a **decision tree to**



predict customer income based on occupation, product, background and various other variables. So In this case, we are predicting values for continuous variables.

We all know that the **Terminal nodes (or leaves) lies at the bottom of the decision tree.** This means that **Decision Trees** are typically drawn upside down such that **leaves are at the the bottom & roots are at the top**

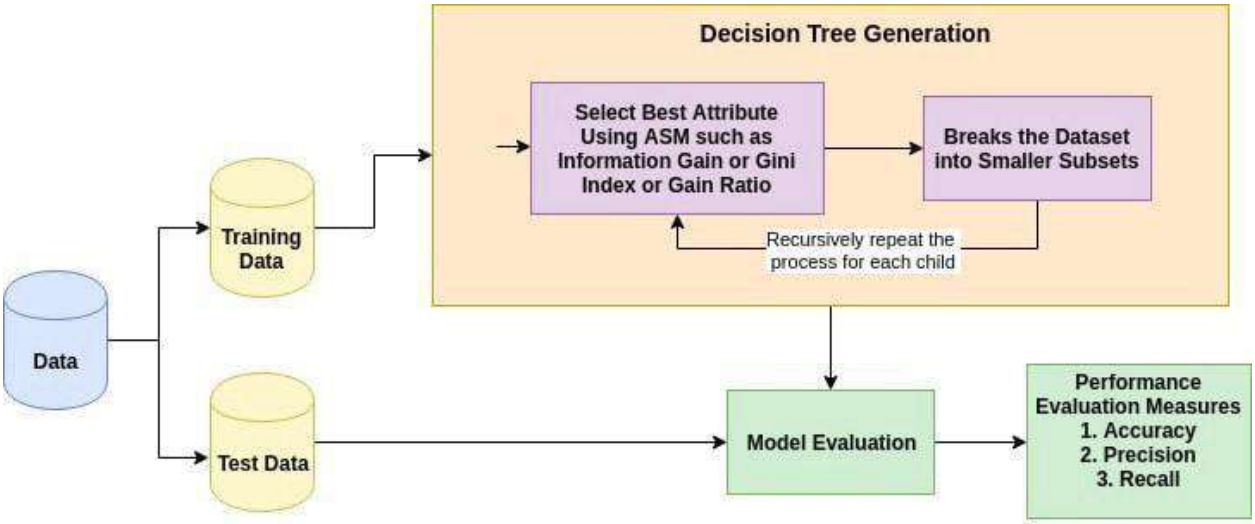


Simple Decision Tree Implementation
Implement the Decision tree.
Steps will also remain the same, which are given below:

- Data Preprocessing 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.

Working on a Decision Tree

1. The root node feature is selected based on the results from the Attribute Selection Measure(ASM).
2. The ASM is repeated until there is a leaf node or a terminal node where it cannot be split into sub-nodes.



What is Attribute Selective Measure(ASM)?

Attribute Subset Selection Measure is a technique used in the data mining process for data reduction. The data reduction is necessary to make better analysis and prediction of the target variable.

The two main ASM techniques are

1. Gini index
2. Information Gain(ID3)

Gini index

The measure of the degree of probability of a particular variable being wrongly classified when it is randomly chosen is called the Gini index or Gini impurity. The data is equally distributed based on the Gini index.

Mathematical Formula :

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

Pi= probability of an object being classified into a particular class.

When you use the Gini index as the criterion for the algorithm to select the feature for the root node.,The feature with the least Gini index is selected.

Information Gain(ID3)

Entropy is the main concept of this algorithm which helps in determining a feature or attribute that gives maximum information about a class is called Information gain or ID3 algorithm. By using this method we can reduce the level of entropy from the root node to the leaf node.

Mathematical Formula :

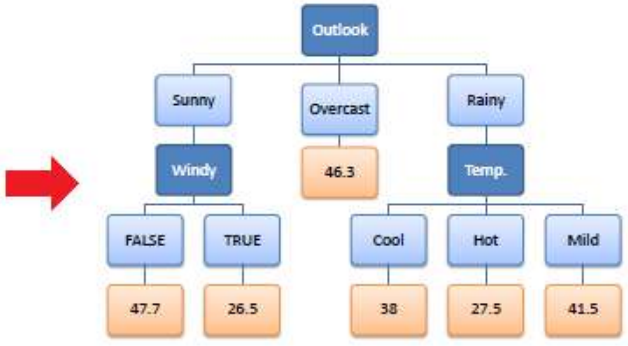
$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

'*p*', denotes the probability of E(S) which denotes the entropy. The feature or attribute with the highest ID3 gain is used as the root for the splitting**.**

Understanding Decision Tree Regression

Decision trees build regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g.,Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

Predictors				Target
Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	25
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



Implementation of Decision Tree Regression

Regression trees are used when the dependent variable is continuous.
For regression trees, the value of terminal nodes is the mean of the observations falling in that region. Therefore, if an unseen data point falls in that region, we predict using the mean value.

Import the libraries

We need to import:

- NumPy
- Pandas
- DecisionTreeRegressor
- train_test_split
- r2_score
- mean squared error
- and Seaborn.

Load the data

Once the libraries are imported, our next step is to load the data, stored here. You can download the data and keep it in your local folder. After that we can use the **read_csv** method of Pandas to load the data into a Pandas data frame **df.Also** shown in the snapshot of data below, the data frame has two columns, x and y. Here x is the feature and y is the label. We're going to predict y using x as an independent variable.

Data pre-processing

Before feeding the data to the tree model, we need to do some pre-processing.
Here, we'll create the x and y variables by taking them from the dataset and using the **train_test_split** function of scikit-learn to split the data into training and test sets.

We also need to reshape the values using the reshape method so that we can pass the data to **train_test_split** in the format required.

Note that the test size of 0.5 indicates we've used 50% of the data for testing. **random_state** ensures reproducibility. For the output of **train_test_split** we get **x_train, x_test, y_train**, and **y_test** values.

Fit the model

We're going to use **x_train** and **y_train** obtained above to train our decision tree regression model. We're using the fit method and passing the parameters as shown below.

Note that the output of this cell is describing a large number of parameters like criteria, max depth, etc for the model. All these parameters are configurable, and you're free to tune them to match your requirements.

Model evaluation

Finally, we need to check to see how well our model is performing on the test data. For this, we evaluate our model by finding the root mean squared error produced by the model.
Mean squared error is a built in function, and we are using NumPy's square root function **(np.sqrt)** on top of it to find the root mean squared error value.