

GeeksforGeeks

A computer science portal for geeks

COURSES

LOGIN

HIRE WITH US

Decision Tree
Introduction with
example

Best Python
libraries for
Machine Learning

Neural Networks |
A beginners guide

ML | Stochastic
Gradient Descent
(SGD)

DBMS | OLAP
Operations

Generative
Adversarial
Network (GAN)

ML | Classification
vs Regression

Game Playing in
Artificial
Intelligence

ML | Mini-Batch
Gradient Descent
with Python

Q-Learning in
Python

Implement your
own
word2vec(skip-
gram) model in
Python

Python | Tokenize
text using TextBlob

Deep Learning |
Introduction to

Long Short Term
Memory

ML | Using SVM to
perform
classification on a
non-linear dataset

ML | Locally
weighted Linear
Regression

ML | Monte Carlo
Tree Search
(MCTS)

Applying
Multinomial Naive
Bayes to NLP
Problems

Data Structure to
Design a special
social network

ML | Natural
Language
Processing using
Deep Learning

MATLAB | Display
histogram of a
grayscale Image

Audio processing
using Pydub and
Google
speechRecognition
API

LSB based Image
steganography
using MATLAB

Elasticsearch
Search Engine | An
introduction

Python | Speech
recognition on
large audio files

Draw Austria flag
using Matlab

Introduction to
Signals and
Systems:

Properties of
systems

Text extraction
from image using
LSB based
steganography

Draw Bangladesh
Flag Using Matlab

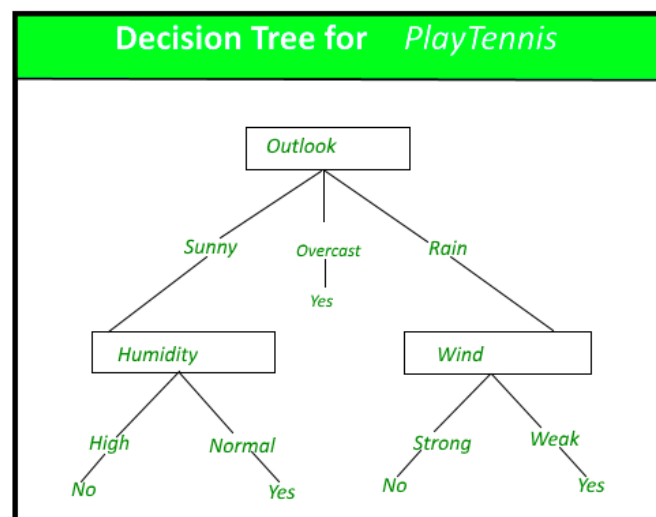
MATLAB | Convert
video into slow
motion

Image Resizing in
Matlab



Decision Tree

Decision Tree : Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



A decision tree for the concept PlayTennis.



GATECS 2020
Blended Course

> Learn from IISC/IIT's Alumni
> 2 LIVE sessions with 2 practice tests per week

Construction of Decision Tree :

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

Decision Tree Representation :

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.

The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf. (in this case Yes or No).

For example, the instance

(Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Strong)



GATECS 2020
Blended Course

> Learn from IISC/IIT's Alumni
> 2 LIVE sessions with
2 practice tests per week

would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.

In other words we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook = Sunny ^ Humidity = Normal) v (Outlook = Overcast) v (Outlook = Rain ^ Wind = Weak)

Strengths and Weakness of Decision Tree approach

The strengths of decision tree methods are:

- Decision trees are able to generate understandable rules.
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

The weaknesses of decision tree methods :

- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
- Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

References :

Machine Learning, Tom Mitchell, McGraw Hill, 1997.

In the next post we will be discussing about ID3 algorithm for the construction of Decision tree given by J. R. Quinlan.

This article is contributed by **Saloni Gupta**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

[Decision Tree Introduction with example](#)

[Decision tree implementation using Python](#)

[Python | Decision Tree Regression using sklearn](#)

[ML | Logistic Regression v/s Decision Tree Classification](#)

[Markov Decision Process](#)

[Maximum sub-tree sum in a Binary Tree such that the sub-tree is also a BST](#)

[Complexity of different operations in Binary tree, Binary Search Tree and AVL tree](#)

[Convert an arbitrary Binary Tree to a tree that holds Children Sum Property](#)

[Given level order traversal of a Binary Tree, check if the Tree is a Min-Heap](#)

[Print Binary Tree levels in sorted order | Set 3 \(Tree given as array\)](#)

[Check if a given Binary Tree is height balanced like a Red-Black Tree](#)

[Convert a given Binary tree to a tree that holds Logical AND property](#)

[Check whether a binary tree is a complete tree or not | Set 2 \(Recursive Solution\)](#)

[Sub-tree with minimum color difference in a 2-coloured tree](#)

[Check if the given binary tree has a sub-tree with equal no of 1's and 0's | Set 2](#)

Article Tags : [Advanced Computer Subject](#) [Machine Learning](#) [Tree](#)

Practice Tags : [Tree](#) [Machine Learning](#)



3

☐ To-do ☐ Done

4

Based on 2 vote(s)

[Feedback/ Suggest Improvement](#)

[Add Notes](#)

[Improve Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

GeeksforGeeks
A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Courses
Company-wise
Topic-wise
How to begin?

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved