Apriori Algorithm:

The Apriori Algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions.

steps for apriori Algorithm:

1) Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

2) Find all the rules of these subsets that 've higher confidence value than the threshold or minimum confidence

3) Take all the supports in the transaction with higher confidence value than the threshold or selected support value.

4) Sort the rules as the decreasing order of lift.

Working of Apriori Algorithm:

Step -1

Create a table that contains support count of each item in the given dataset.

Itemset	support_count
A	6
B	7
C	5
D	2
E	1

step-2:

In this step, we'll generate C2 with the help of L1. In C2, we'll create the pair of the itemset of L1 in the form of subsets.

Itemset	support_count
A, B	4
A, C	4
A, D	1
B, C	4
B, D	2
C, D	0

Again, we need to compare the C2 support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table.

Itemset	support_count
A, B	4
A, C	4
B, C	4
B, D	2

step - 3 :

will repeat the same two processes, but now we'll form the table with subsets of three itemsets together, and will calculate the support count from dataset.

Itemset	support_count
A, B, C	2
B, C, D	1
A, C, D	0
A, B, D	0

step-4:

To generate the association rules, first, we will create a new table with the possible rules from the occured combination {A, B, C}. For all the rules, we'll calculate the confidence value for all rules, we'll exclude the rules that have less confidence than the minimum threshold (50%).

Rules	support	Confidence.
A^B→C	2	sup{(A^B)^C}/sup(A^B) = 2/4 = 0.5 = 50%.
B^C→A	2	sup{(B^C)^A}/sup(B^C) = 2/4 = 0.5 = 50%.
A^C→B	2	sup{(A^C)^B}/(sup(A^C) = 2/4 = 0.5 = 50%.
C→A^B	2	sup{C^(A^B)}/sup(C) = 2/5 = 0.4 = 40%.
A→B^C	2	sup{A^(B^C)}/(sup(A) = 2/6 = 0.33 = 33.33%.

Rules	support	Confidence.
B → B^C	2	sup{(B^(B^C)} / sup(B) = 2/7 = 0.28 = 28%.

As the given threshold or minimum confidence is 50%, so the first three rules A^B → C, B^C → A, and A^C → B can be considered as the strong association rules for the given problem.

Ad. of Apriori Algorithm:

Easy to understand Algorithm.

Join and prune steps of the Algorithm can be easily implemented on large datasets.

VIJAY PRAKASH M

20CSEA 30

Decision Tree

16 - 09 - 2021.

Introduction:

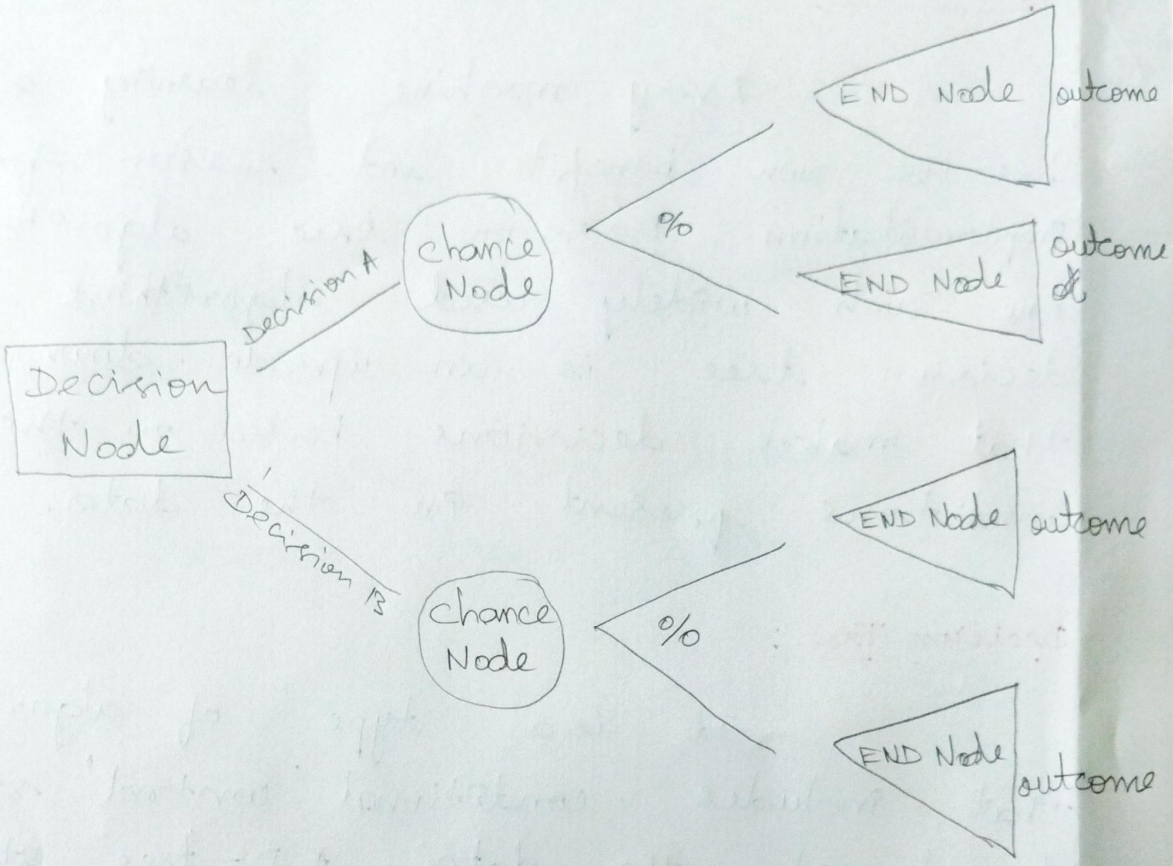
Every machine learning algorithms has its own benefits and reason for implementation. Decision tree algorithms is one such widely used algorithms. A decision tree is an upside-down tree that makes decisions based on the conditions present in the data.

Decision Tree:

It is a type of algorithm that includes conditional 'control' statements to classify the data. A D-tree starts at a single point which then branches in two or more directions.

Decision trees are extremely useful for data analytics, because they break down complex data into more manageable parts. They're often used in these fields for prediction analysis, data classification and regression.

Parts of D-tree:



Decision Node Represents decision.
Chance Nodes Represents probability or uncertainty.
End Nodes Representing an outcome.

Ad. of D-Tree:

Good for interpreting data in a highly visual way.

Easy to define rules.

requires minimal preparation or data cleaning before use.

Disadd. of D-tree:

They're not well-suited to continuous variables.

Overfitting can become a problem if a decision tree's design is too complex.

Why D-tree:

D-tree provides an effective method of decision Making. Because they clearly layout the problem so that all options can be challenged. Allows us to analyze fully the possible consequences of a decision.

Where it used:

D-trees are used for handling non-linear data sets effectively. The decision tree tool is used in real life in many areas like, law, business, etc,..

Conclusion:

As a goal of a D-tree is that it makes the optimal choice at the end of each node it needs an algorithm that is capable of doing just that.

Entropy formula:

$$\sum_{i=1}^{k} p(\text{value } i) \cdot \log_2 (p(\text{value } i))$$

From the dataset:

Out of 14 instances we've 9 yes and 5 NO's.

$$E(s) = p(\text{yes}) \log_2 p(\text{yes}) - p(\text{No}) \log_2 p(\text{No})$$

$$E(s) = (9/14) * \log_2 9/4 - (5/14) * \log_2 5/14$$

$$= (-0.642) * (-(0.639)) - (0.37) * (-1.486)$$

$$E(s) = 0.94$$

Outlook $\Rightarrow$ sunny:

$$E(\text{outlook} = \text{sunny}) = -p(\text{yes}) \log_2 P(\text{yes}) - P(\text{No}) \log_2 P(\text{No})$$

$$= -0.4 \log_2 2/5 - 3/5 \log_2 (3/5)$$

$$= 0.328 + 0.438$$

$$= 0.97$$

E (outlook = overcast) = $(-4/4) * \log_2 4/4 + (-0/4) \log_2 0/4$

$= (-1) * \log_2 1 * \log_2 0$

$= (-) * 0 + 0$

$= 0$

E(outlook = rainy) = $(-3/5) \log_2 3/5 + (-2/5) \log_2 2/5$

$= (-0.6) * (-0.73) + (-0.4) * (-1.32)$

$= 0.438 + 0.528$

$= 0.97.$

I (outlook) = $5/14 \times 0.97 + 0 + 5/14 \times 0.97$

$= 0.693$

temp:

E(temp = hot) = $(-2/4) \log_2 2/4 + (-2/4) \log_2 2/4$

$= (-0.5) + (-1.0) + (-0.5) * (1.0)$

$= 1$

E(temp = mild) = $(-4/6) \log_2 4/6 + (-2/6) \log_2 2/6$

$= (-0.67) * (0.58) + (-0.33) * (-1.6)$

$= 0.389 + 0.528$

$= 0.917$

$E(\text{temp} = \text{cool}) : (-3/4)\log_2 3/4 + (-1/4)\log_2 1/4$

$= (-0.75) * (-0.415) + (-0.25) * (-2.0)$

$= 0.311 + 0.5$

$= 0.811.$

$I(\text{temp}) = 4/14 \times 1 + 6/14 \times 0.917 + 4/14 \times 0.811$

$= 0.285 + 0.392 + 0.231$

$= 0.908.$

Humidity :

$E(\text{humidity} = \text{high}) = (-3/7)\log_2 3/7 + (-4/7)\log_2 4/7$

$= (-0.429) * (-1.22) + (-0.571) * (0.808)$

$= 0.523 + 0.461$

$= 0.984.$

$E(\text{humidity} = \text{normal}) = (-6/7)\log_2 6/7 + (-1/7)\log_2 1/7$

$= 0.190 + 0.399$

$= 0.589.$

$I(\text{humidity}) = 7/14 * 0.984 + 7/14 * 0.589$

$= 0.492 + 0.294$

$= 0.786.$

Windy:

$E(\text{windy} = \text{False}) = (-6/8)\log_2 6/8 + (-2/8)\log_2 2/8$

$= 0.311 + 0.5$

$= 0.811.$

$E(\text{windy} = \text{True}) = (-3/6)\log_2 3/6 + (-3/6)\log_2 3/6$

$= 0.5 + 0.5$

$= 1$

$I(\text{windy}) = 8/14 * 0.811 + 6/14 * 1$

$= 0.571 * 0.811 + 0.42$

$= 0.892.$

outlook:

$\text{Gain} = 0.94 - 0.693$

$= 0.247$

Temperature:

$\text{Gain} = 0.94 - 0.908$

$= 0.032$

Humidity :

$$\text{gain} = 0.94 - 0.788$$
$$= 0.154$$

Windy :

$$\text{gain} = 0.94 - 0.892$$
$$= 0.048$$

Max. gain = 0.247.

So, outlook is our Root Node.