A Step by Step ID3 Decision Tree Example

November 20, 2017 / Machine Learning

Decision tree algorithms transfom raw data to rule based decision making trees. Herein, ID3 is one of the most common decision tree algorithm. Firstly, It was introduced in 1986 and it is acronym of **Iterative Dichotomiser**.

First of all, dichotomisation means dividing into two completely opposite things. That's why, the algorithm iteratively divides attributes into two groups which are the most dominant attribute and others to construct a tree. Then, it calculates the entropy and information gains of each atrribute. In this way, the most dominant attribute can be founded. After then, the most dominant one is put on the tree as decision node. Thereafter, entropy and gain scores would be calculated again among the other attributes. Thus, the next most dominant attribute is found. Finally, this procedure continues until reaching a decision for that branch. That’s why, it is called Iterative Dichotomiser. So, we’ll mention the algorithm step by step in this post.

Sandra Bullock, Premonition (2007)

For instance, the following table informs about decision making factors to play tennis at outside for previous 14 days.

Day	**Outlook**	**Temp.**	**Humidity**	**Wind**	**Decision**
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

We can summarize the ID3 algorithm as illustrated below

$\text{Entropy}(S) = \sum - p(I) \cdot \log_2 p(I)$

$\text{Gain}(S, A) = \text{Entropy}(S) - \sum [\, p(S|A) \cdot \text{Entropy}(S|A) \,]$

These formulas might confuse your mind. Practicing will make it understandable.

Entropy

We need to calculate the entropy first. Decision column consists of 14 instances and includes two labels: yes and no. There are 9 decisions labeled yes, and 5 decisions labeled no.

$\text{Entropy}(\text{Decision}) = - p(\text{Yes}) \cdot \log_2 p(\text{Yes}) - p(\text{No}) \cdot \log_2 p(\text{No})$

$\text{Entropy}(\text{Decision}) = - (9/14) \cdot \log_2(9/14) - (5/14) \cdot \log_2(5/14) = 0.940$

Now, we need to find the most dominant factor for decisioning.

Wind factor on decision

$\text{Gain}(\text{Decision}, \text{Wind}) = \text{Entropy}(\text{Decision}) - \sum [\, p(\text{Decision}|\text{Wind}) \cdot \text{Entropy}(\text{Decision}|\text{Wind}) \,]$

Wind attribute has two labels: weak and strong. We would reflect it to the formula.

$\text{Gain}(\text{Decision}, \text{Wind}) = \text{Entropy}(\text{Decision}) - [\, p(\text{Decision}|\text{Wind=Weak}) \cdot \text{Entropy}(\text{Decision}|\text{Wind=Weak}) \,] - [\, p(\text{Decision}|\text{Wind=Strong}) \cdot \text{Entropy}(\text{Decision}|\text{Wind=Strong}) \,]$

Now, we need to calculate (Decision|Wind=Weak) and (Decision|Wind=Strong) respectively.

Weak wind factor on decision

Day	**Outlook**	**Temp.**	**Humidity**	**Wind**	**Decision**

1	Sunny	Hot	High	Weak	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
13	Overcast	Hot	Normal	Weak	Yes

There are 8 instances for weak wind. Decision of 2 items are no and 6 items are yes as illustrated below.

1- Entropy(Decision|Wind=Weak) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)

2- Entropy(Decision|Wind=Weak) = – (2/8) . $\log_2$(2/8) – (6/8) . $\log_2$(6/8) = 0.811

Notice that if the number of instances of a class were 0 and total number of instances were n, then we need to calculate -(0/n) . $\log_2$(0/n). Here, log(0) would be equal to -∞, and we cannot calculate 0 times ∞. This is a special case often appears in decision tree applications. Even though compilers cannot compute this operation, we can compute it with calculus. If you wonder how to compute this equation, please read this post.

Strong wind factor on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
2	Sunny	Hot	High	Strong	No

6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
14	Rain	Mild	High	Strong	No

Here, there are 6 instances for strong wind. Decision is divided into two equal parts.

1- Entropy(Decision|Wind=Strong) = – p(No) . $\log_2$p(No) – p(Yes) . $\log_2$p(Yes)

2- Entropy(Decision|Wind=Strong) = – (3/6) . $\log_2$(3/6) – (3/6) . $\log_2$(3/6) = 1

Now, we can turn back to Gain(Decision, Wind) equation.

Gain(Decision, Wind) = Entropy(Decision) – [p(Decision|Wind=Weak) . Entropy (Decision|Wind=Weak)] – [p(Decision|Wind=Strong) . Entropy (Decision|Wind=Strong)] = 0.940 – [(8/14) . 0.811] – [(6/14). 1] = 0.048

Calculations for wind column is over. Now, we need to apply same calculations for other columns to find the most dominant factor on decision.

Other factors on decision

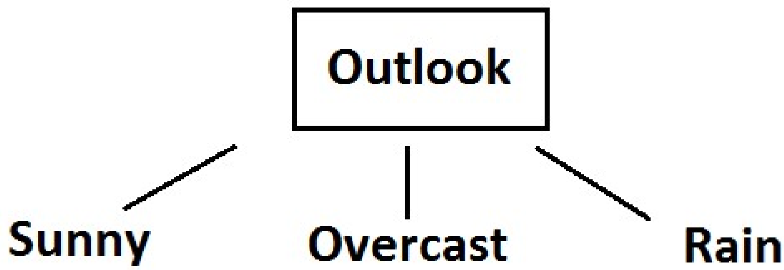
We have applied similar calculation on the other columns.

1- Gain(Decision, Outlook) = 0.246

2- Gain(Decision, Temperature) = 0.029

3- Gain(Decision, Humidity) = 0.151

As seen, outlook factor on decision produces the highest score. That's why, outlook decision will appear in the root node of the tree.



Root decision on the tree

Now, we need to test dataset for custom subsets of outlook attribute.

Overcast outlook on decision

Basically, decision will always be yes if outlook were overcast.

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Sunny outlook on decision

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No

8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Here, there are 5 instances for sunny outlook. Decision would be probably 3/5 percent no, 2/5 percent yes.

1- Gain(Outlook=Sunny|Temperature) = 0.570

2- Gain(Outlook=Sunny|Humidity) = 0.970

3- Gain(Outlook=Sunny|Wind) = 0.019

Now, humidity is the decision because it produces the highest score if outlook were sunny.

At this point, decision will always be no if humidity were high.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

On the other hand, decision will always be yes if humidity were normal

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Finally, it means that we need to check the humidity and decide if outlook were sunny.

Rain outlook on decision

Day	**Outlook**	**Temp.**	**Humidity**	**Wind**	**Decision**
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

1- Gain(Outlook=Rain | Temperature)

2- Gain(Outlook=Rain | Humidity)

3- Gain(Outlook=Rain | Wind)

Here, wind produces the highest score if outlook were rain. That's why, we need to check wind attribute in 2nd level if outlook were rain.

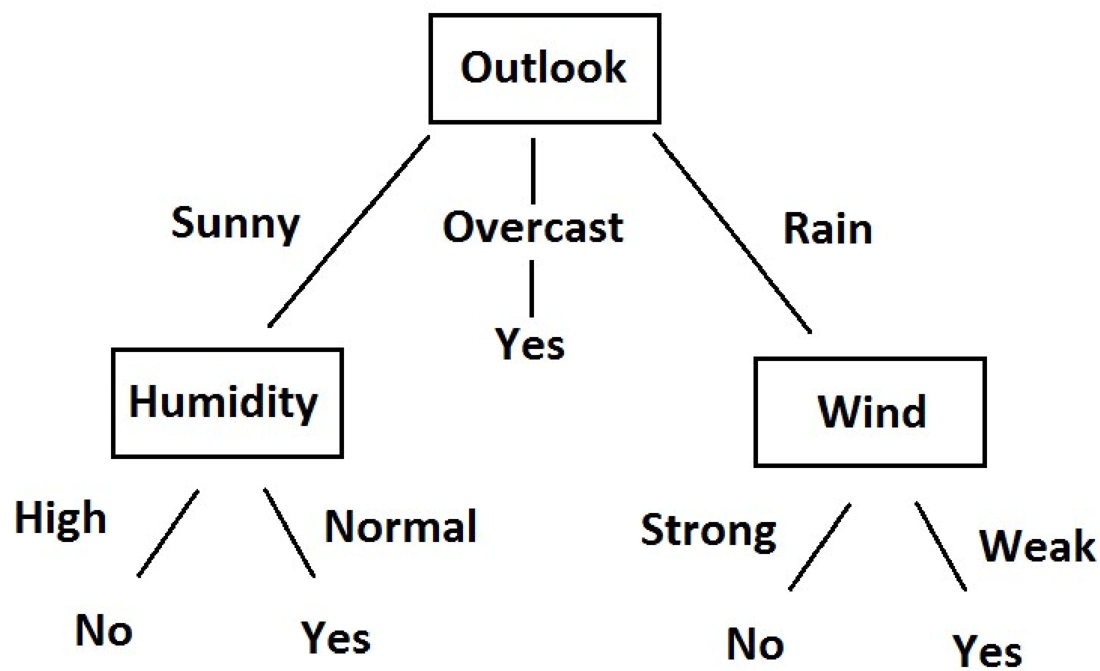
So, it is revealed that decision will always be yes if wind were weak and outlook were rain.

Day	**Outlook**	**Temp.**	**Humidity**	**Wind**	**Decision**
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

What’s more, decision will be always no if wind were strong and outlook were rain.

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

So, decision tree construction is over. We can use the following rules for decisioning.

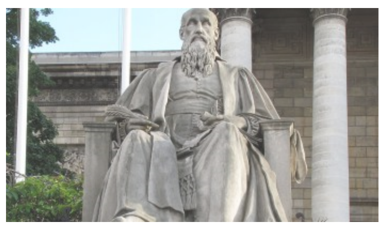


Final version of decision tree

So, decision tree algorithms transfrom the raw data into rule based mechanism. In this post, we have mentioned one of the most common decision tree algorithm named as ID3. They can use nominal attributes whereas most of common machine learning algorithms cannot. However, it is required to transform numeric attributes to nominal in ID3. Besides, its evolved version C4.5 exists which can handle nominal data. Even though decision tree algorithms are powerful, they have long training time. On the other hand, they tend to fall over-fitting. Besides, they have evolved versions named random forests which tend not to fall over-fitting issue and have shorter training times.

Related

A Step By Step C4.5 Decision Tree Example



Indeterminate Forms and L'Hospital's Rule in Decision Trees

How Random Forests Can Keep You From Decision Tree

#decision tree, #id3, #random forest

« Previous / Next »

9 Comments

Workneh

June 6, 2018 at 9:00 pm

Thank you dear! It’s very helpful.

Reply

OGBONNA VITALIS

June 7, 2018 at 5:27 am

Thanks alot, very nice of

Reply

mr chidubem okafor

June 7, 2018 at 1:18 pm

very cool

Reply

jane doe

July 4, 2018 at 7:34 am

Instead of (Outlook=Sunny|Temperature)gain, shouldn’t it be Gain (Outlook=Sunny|Temperature)?

Reply

Sefik Serengil

July 5, 2018 at 1:20 pm

Right, I fixed the terms. Thanks.

Reply

Zubby zelletan
July 19, 2018 at 5:27 am

Thanks alot, the explanation was helpful.

Reply

Mustajab Hussain
October 28, 2018 at 9:38 am

Please give me the python code of this algorithm

Reply

Sefik Serengil
October 28, 2018 at 9:40 am

You can find it in my GitHub repo. Link is https://github.com/serengil/decision-trees-for-ml

Reply

Mustajab Hussain
October 28, 2018 at 9:54 am

Thanks Sefik Serengil. You are such a helping man

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

ex: jane doe

Email *

ex: janedoe@gmail.com

Website

ex: http://janedoe.wordpress.com

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

Submit

You can follow this blog and receive notifications for new posts

Email *

Subscribe