

Decision Tree - Regression

Decision tree builds 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.



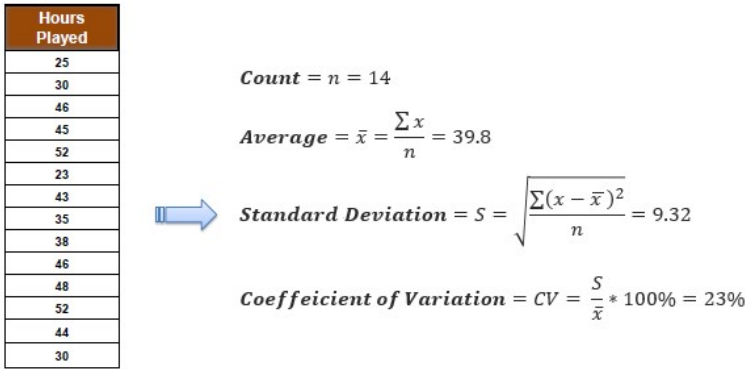
Decision Tree Algorithm

The core algorithm for building decision trees called **ID3** by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. The ID3 algorithm can be used to construct a decision tree for regression by replacing Information Gain with *Standard Deviation Reduction*.

Standard Deviation

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). We use standard deviation to calculate the homogeneity of a numerical sample. If the numerical sample is completely homogeneous its standard deviation is zero.

a) Standard deviation for **one** attribute:



- Standard Deviation (**S**) is for tree building (branching).
- Coefficient of Deviation (**CV**) is used to decide when to stop branching. We can use Count (**n**) as well.
- Average (**Avg**) is the value in the leaf nodes.

b) Standard deviation for **two** attributes (target and predictor):

$$S(T, X) = \sum_{c \in X} P(c) S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14

$$\begin{aligned} S(\text{Hours, Outlook}) &= P(\text{Sunny}) * S(\text{Sunny}) + P(\text{Overcast}) * S(\text{Overcast}) + P(\text{Rainy}) * S(\text{Rainy}) \\ &= (4/14) * 3.49 + (5/14) * 7.78 + (5/14) * 10.87 \\ &= 7.66 \end{aligned}$$

Standard Deviation Reduction

The standard deviation reduction is based on the decrease in standard deviation after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest standard deviation reduction (i.e., the most homogeneous branches).



Adobe Creative Cloud for Teams starting at \$29.99 per month.

ADS VIA CARBON

Step 1: The standard deviation of the target is calculated.

Standard deviation (Hours Played) = 9.32

Step 2: The dataset is then split on the different attributes. The standard deviation for each branch is calculated. The resulting standard deviation is subtracted from the standard deviation before the split. The result is the standard deviation reduction.

		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR=0.17		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

$SDR(T, X) = S(T) - S(T, X)$

$SDR(\text{Hours}, \text{Outlook}) = S(\text{Hours}) - S(\text{Hours}, \text{Outlook})$
 $= 9.32 - 7.66 = 1.66$

Step 3: The attribute with the largest standard deviation reduction is chosen for the decision node.

		Hours Played (StDev)
★ Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

Step 4a: The dataset is divided based on the values of the selected attribute. This process is run recursively on the non-leaf branches, until all data is processed.

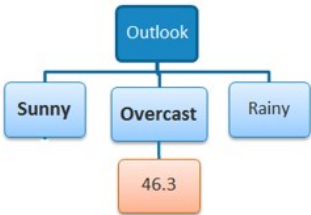
		Outlook	Temp	Humidity	Windy	Hours Played
Outlook	Sunny		Mild	High	FALSE	45
			Cool	Normal	FALSE	52
			Cool	Normal	TRUE	23
			Mild	Normal	FALSE	46
			Mild	High	TRUE	30
	Overcast		Hot	High	FALSE	46
			Cool	Normal	TRUE	43
			Mild	High	TRUE	52
			Hot	Normal	FALSE	44
	Rainy		Hot	High	FALSE	25
			Hot	High	TRUE	30
			Mild	High	FALSE	35
			Cool	Normal	FALSE	38
			Mild	Normal	TRUE	48

In practice, we need some termination criteria. For example, when coefficient of deviation (CV) for a branch becomes smaller than a certain threshold (e.g., 10%) and/or when too few instances (n) remain in the branch (e.g., 3).

Step 4b: "Overcast" subset does not need any further splitting because its CV (8%) is less than the threshold (10%). The related leaf node gets the average of the "Overcast" subset.

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



Step 4c: However, the "Sunny" branch has an CV (28%) more than the threshold (10%) which needs further splitting. We select "Windy" as the best best node after "Outlook" because it has the largest SDR.

Outlook - Sunny

		Hours Played (StDev)	Count
Temp	Cool	14.50	2
	Mild	7.32	3

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Cool	Normal	TRUE	23
Mild	Normal	FALSE	46
Mild	High	TRUE	30
			$S = 10.87$
			$AVG = 39.2$
			$CV = 28\%$

$$SDR = 10.87 - ((2/5) * 14.5 + (3/5) * 7.32) = 0.678$$

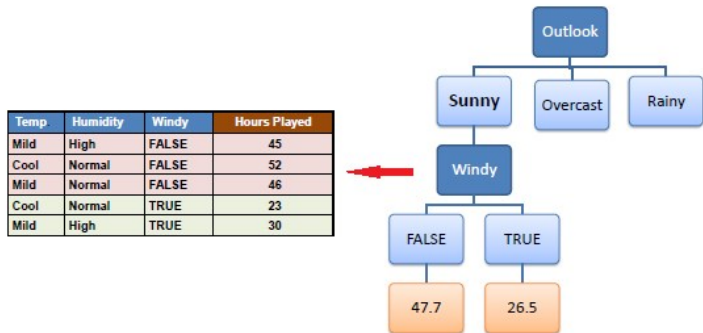
		Hours Played (StDev)	Count
Humidity	High	7.50	2
	Normal	12.50	3

$$SDR = 10.87 - ((2/5) * 7.5 + (3/5) * 12.5) = 0.370$$

		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

$$SDR = 10.87 - ((3/5) * 3.09 + (2/5) * 3.5) = 7.62$$

Because the number of data points for both branches (FALSE and TRUE) is equal or less than 3 we stop further branching and assign the average of each branch to the related leaf node.



Step 4d: Moreover, the "rainy" branch has an CV (22%) which is more than the threshold (10%). This branch needs further splitting. We select "Windy" as the best best node because it has the largest SDR.

Outlook - Rainy

Temp	Humidity	Windy	Hours Played
Hot	High	FALSE	25
Hot	High	TRUE	30
Mild	High	FALSE	35
Cool	Normal	FALSE	38
Mild	Normal	TRUE	48
			$S = 7.78$
			$AVG = 35.2$
			$CV = 22\%$

		Hours Played (StDev)	Count
Temp	Cool	0	1
	Hot	2.5	2
	Mild	6.5	2

$$SDR = 7.78 - ((1/5) * 0 + (2/5) * 2.5 + (2/5) * 6.5) = 4.18$$

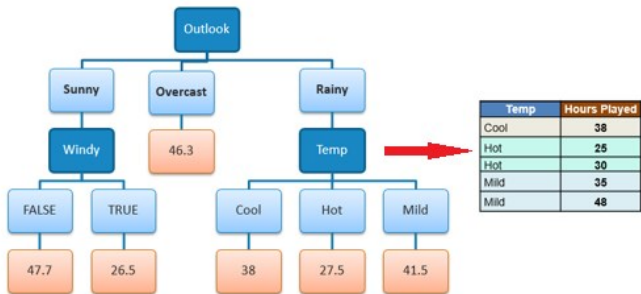
		Hours Played (StDev)	Count
Humidity	High	4.1	3
	Normal	5.0	2

$$SDR = 7.78 - ((3/5) * 4.1 + (2/5) * 5.0) = 3.32$$

		Hours Played (StDev)	Count
Windy	False	5.6	3
	True	9.0	2

$$SDR = 7.78 - ((3/5) * 5.6 + (2/5) * 9.0) = 0.82$$

Because the number of data points for all three branches (Cool, Hot and Mild) is equal or less than 3 we stop further branching and assign the average of each branch to the related leaf node.



When the number of instances is more than one at a leaf node we calculate the average as the final value for the target.

Exercise

Try to invent a new algorithm to construct a decision tree from data using MLR instead of average at the leaf node.