Decision Trees are one of the best known supervised **classification methods**. "*A decision tree is a way of representing knowledge obtained in the inductive learning process. The space is split using a set of conditions, and the resulting structure is the tree*"
A tree is composed of nodes, and those nodes are chosen looking for the **optimum split** of the features. For that purpose, different criteria exist. In the decision tree Python implementation of the scikit-learn library, this is made by the parameter '*criterion*'. This parameter is the function used to measure the quality of a split and it allows users to choose between '*gini*' or '*entropy*'.
How does each criterion find the optimum split? And, what are the differences between both of them? In this post, we are going to answer these questions. First, explaining **gini** and **entropy** criteria and their differences, and then, a practical example that compares both of them is presented.
Moreover, if you are interested in decision trees, this **post** about tree ensembles may be of your interest.

Gini

The **gini impurity** is calculated using the following formula:

$$GiniIndex = 1 - \sum_j p_j^2$$

The gini impurity measures the frequency at which any element of the dataset will be mislabelled when it is randomly labelled.

The minimum value of the Gini Index is 0. This happens when the node is **pure**, this means that all the contained elements in the node are of one unique class. Therefore, this node will not be split again. Thus, the optimum split is chosen by the features with less Gini Index. Moreover, it gets the maximum value when the probability of the two classes are the same.

$$Gini_{min} = 1 - (1^2) = 0$$

$$Gini_{max} = 1 - (0.5^2 + 0.5^2) = 0.5$$

Entropy

The **entropy** is calculated using the following formula:

$$Entropy = -\sum_j p_j \cdot log_2 \cdot p_j$$

Where, as before, p_j is the probability of class j.

$$Entropy_{min} = -1 \cdot log_2(1) = 0$$

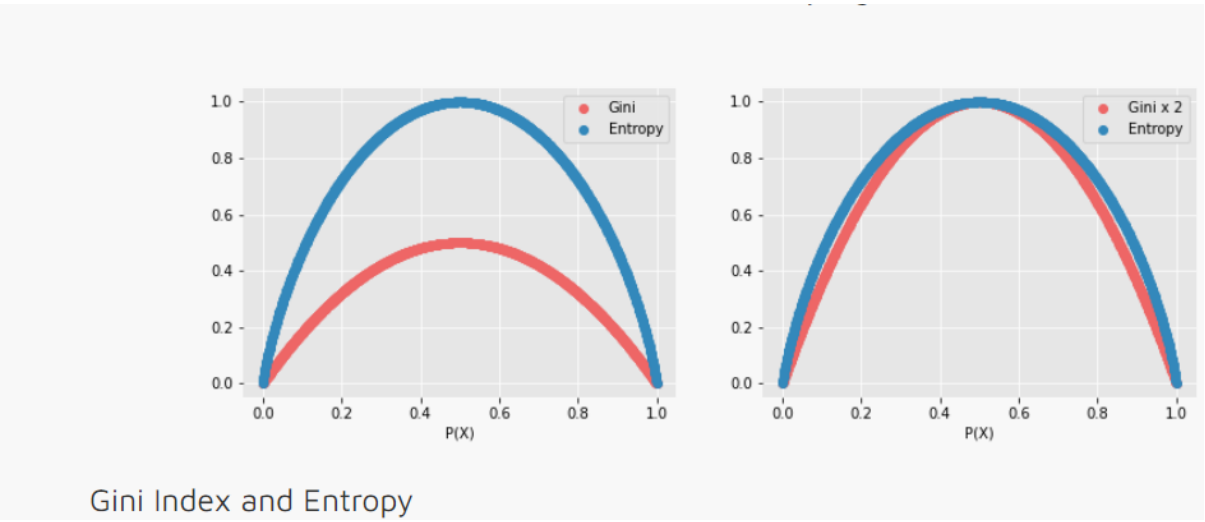
$$Entropy_{max} = -0.5 \cdot log_2(0.5) - 0.5 \cdot log_2(0.5) = 1$$

Gini vs Entropy

The Gini Index and the Entropy have two main differences:

- Gini Index has values inside the interval [0, 0.5] whereas the interval of the Entropy is [0, 1]. In the following figure, both of them are represented. The gini index has also been represented multiplied by two to see concretely the differences between them, which are not very significant.

$$Information\ Gain = Entropy_{\ parent} - Entropy_{\ children}$$



Gini Index and Entropy

Gini Index and Entropy

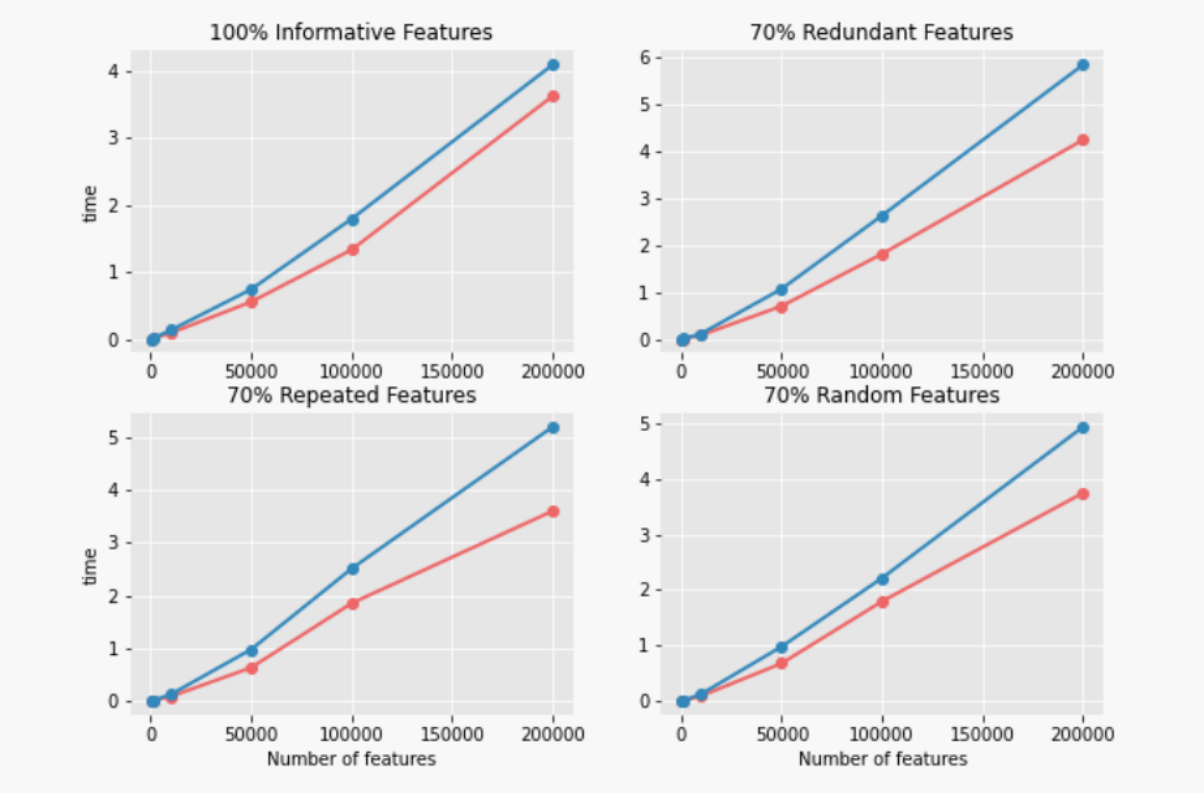
- Computationally, entropy is more complex since it makes use of **logarithms** and consequently, the calculation of the Gini Index will be faster.

Therefore, we are going to analyze the impact on the training time when using one criterion or the other. For that purpose, different synthetic datasets have been generated. All these datasets have 10 features and they can be grouped into 4 groups, depending on whether the features are informative, redundant, repeated, or random:

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Informative	100%	30%	30%	30%
Redundant	0	70%	0	0
Repeated	0	0	70%	0
Random	0	0	0	70%

In this way, we can analyze the impact on the **training time**. Moreover, each group is composed of 5 datasets, where the number of samples varies (100, 1.000, 10.000, 100.000, and 200.000).

In the following graphs, the x-axis is the number of samples of the dataset and the y-axis is the training time.



As can be seen, the training time when using the Entropy criterion is much higher. Furthermore, the impact of the nature of the features is quite noticeable. Redundant features are the most affected ones, making the training time 50% longer than when using informative features. Whereas, the use of random features or repeated features have a similar impact. The differences in training time are more noticeable in larger datasets.

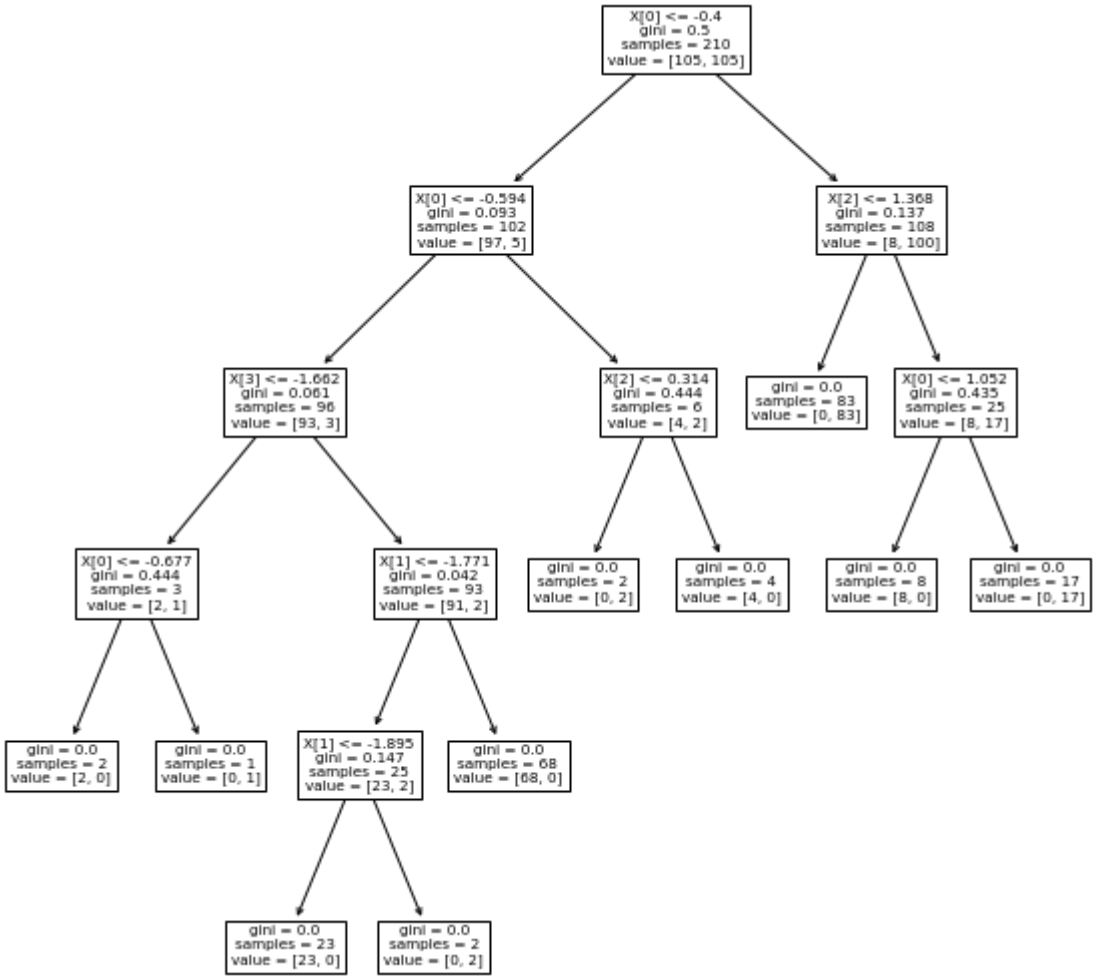
Results

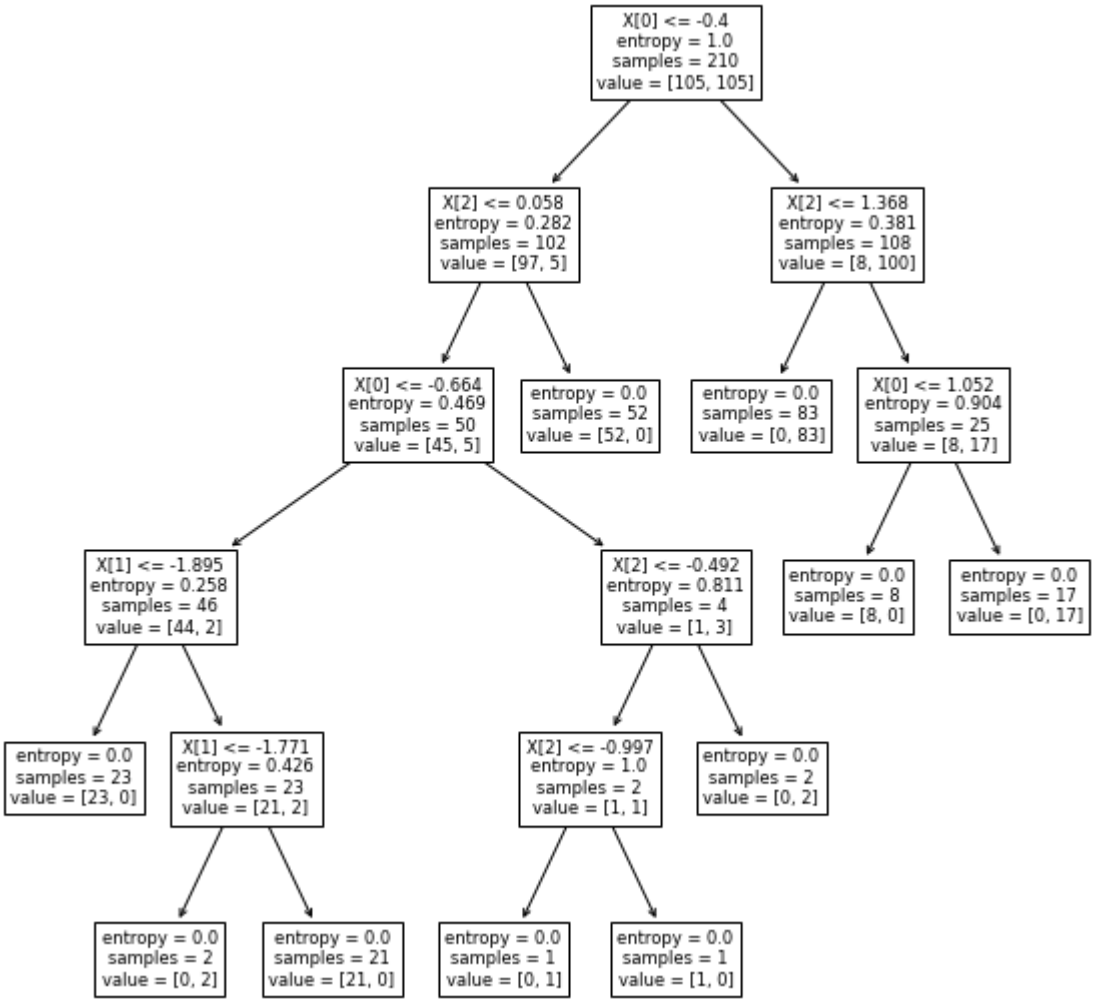
Besides, we are also going to compare the obtained results with both criteria. For that purpose, we are going to use the datasets used in the training time analysis, specifically the ones with 1000 samples. Furthermore, **cross-validation** has also been used with k=3. The following table shows the obtained results, these being the **F-Score**.

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Gini	0.8619 ± 0.0088	0.8148 ± 0.0030	0.9350 ± 0.0146	0.8481± 0.0029
Entropy	0.8659 ± 0.0205	0.8119 ± 0.0223	0.9390 ± 0.0146	0.8583 ± 0.0067

As can be seen, the results are very **similar**, being the ones where the entropy criterion is used slightly better.
Finally, if we compare the structure of the trees, we can see that they are different. For that purpose, we have created a new synthetic dataset with 400 samples and 4 features. The obtained results are an **F-Score** of 0.93 for both criteria but the resulting tree is different. In addition, the first split in the two trees is the same as the branch on the right of the tree, however, the rest of the tree is different.

Although the trees are not equal, the obtained result is practically identical.


X[0] <= -0.4
gini = 0.5
samples = 210
value = [105, 105]
X[0] <= -0.594
gini = 0.093
samples = 102
value = [97, 5]
X[2] <= 1.368
gini = 0.137
samples = 108
value = [8, 100]
X[3] <= -1.662
gini = 0.061
samples = 96
value = [93, 3]
X[2] <= 0.314
gini = 0.444
samples = 6
value = [4, 2]
gini = 0.0
samples = 83
value = [0, 83]
X[0] <= 1.052
gini = 0.435
samples = 25
value = [8, 17]
X[0] <= -0.677
gini = 0.444
samples = 3
value = [2, 1]
X[1] <= -1.771
gini = 0.042
samples = 93
value = [91, 2]
gini = 0.0
samples = 2
value = [0, 2]
gini = 0.0
samples = 4
value = [4, 0]
gini = 0.0
samples = 8
value = [8, 0]
gini = 0.0
samples = 17
value = [0, 17]
gini = 0.0
samples = 2
value = [2, 0]
gini = 0.0
samples = 1
value = [0, 1]
X[1] <= -1.895
gini = 0.147
samples = 25
value = [23, 2]
gini = 0.0
samples = 68
value = [68, 0]
gini = 0.0
samples = 23
value = [23, 0]
gini = 0.0
samples = 2
value = [0, 2]



Terminologies used in decision tree

A Step By Step Regression Tree Example

CART---→ classification and regression tree

Decision trees are powerful way to classify problems. On the other hand, they can be adapted into regression problems, too. Decision trees which built for a

data set where the target column could be real number are called **regression trees**.
All regression model look for the feature offering the highest information gain. Then, they add a decision rule for the found feature and build an another decision tree for the sub data set recursively until they reached a decision.

Besides, regular decision tree, algorithms are designed to create branches for categorical features. Still, we can build trees with continuous and numerical features. The trick is here that we will convert continuous features into categorical. We will split the numerical feature where it offers the highest information gain.

Objective

Decision rules will be found based on standard deviations.

Data set

In the following data set, the target column is number of golf players and it stores real numbers. We have counted the number of instances for each class when the target was nominal(yes/no). we can create branches based on the number of instances for true decisions and false decisions. Here, we cannot count the target values because it is continuous. Instead of counting, we can handle regression problems by switching the metric to standard deviation.

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52

6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30

Standard deviation

Golf players = {25, 30, 46, 45, 52, 23, 43, 35, 38, 46, 48, 52, 44, 30}

Average of golf players = (25 + 30 + 46 + 45 + 52 + 23 + 43 + 35 + 38 + 46 + 48 + 52 + 44 + 30
)/14 = 39.78

Standard deviation of golf players = $\sqrt{[((25 – 39.78)^2 + (30 – 39.78)^2 + (46 – 39.78)^2 + … + (30 – 39.78)^2)/14]}$ = 9.32

Outlook

Outlook can be sunny, overcast and rain. We need to calculate standard deviation of golf players for all of these outlook candidates.

Sunny outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Average of golf players for sunny outlook = (25+30+35+38+48)/5 = 35.2

Standard deviation of golf players for sunny outlook = $\sqrt{(((25 - 35.2)^2 + (30 - 35.2)^2 + \dots)/5)} = 7.78$

Overcast outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

Golf players for overcast outlook = {46, 43, 52, 44}

Average of golf players for overcast outlook = (46 + 43 + 52 + 44)/4 = 46.25

Standard deviation of golf players for overcast outlook = $\sqrt{(((46-46.25)^2+(43-46.25)^2+\ldots)}$= 3.49

Rainy outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Player
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

Golf players for overcast outlook = {45, 52, 23, 46, 30}

Average of golf players for overcast outlook = (45+52+23+46+30)/5 = 39.2

Standard deviation of golf players for rainy outlook = $\sqrt{(((45 – 39.2)^2+(52 – 39.2)^2+\ldots)/5)}$=10.87

Summarizing standard deviations for the outlook feature

Outlook	Stdev of Golf Players	Instances
Overcast	3.49	4
Rain	10.87	5
Sunny	7.78	5

Weighted standard deviation for outlook = (4/14)x3.49 + (5/14)x10.87 + (5/14)x7.78 = 7.66

You might remember that we have calculated the global standard deviation of golf players 9.32 in previous steps. Standard deviation reduction is difference of the global standard deviation and standard deviation for current feature. In this way, maximized standard deviation reduction will be the decision node.

Standard deviation reduction for outlook = 9.32 – 7.66 = 1.66

Temperature

Temperature can be hot, cool or mild. We will calculate standard deviations for those candidates.

Hot temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46

13	Overcast	Hot	Normal	Weak	44

Golf players for hot temperature = {25, 30, 46, 44}

Standard deviation of golf players for hot temperature = 8.95

Cool temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
9	Sunny	Cool	Normal	Weak	38

Golf players for cool temperature = {52, 23, 43, 38}

Standard deviation of golf players for cool temperature = 10.51

Mild temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
8	Sunny	Mild	High	Weak	35
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48

12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for mild temperature = {45, 35, 46, 48, 52, 30}

Standard deviation of golf players for mild temperature = 7.65

Summarizing standard deviations for temperature feature

Temperature	Stdev of Golf Players	Instances
Hot	8.95	4
Cool	10.51	4
Mild	7.65	6

Weighted standard deviation for temperature = (4/14)x8.95 + (4/14)x10.51 + (6/14)x7.65 = 8.84

Standard deviation reduction for temperature = 9.32 – 8.84 = 0.47

Humidity

Humidity is a binary class. It can either be normal or high.

High humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Player
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
8	Sunny	Mild	High	Weak	35
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Day	Outlook	Temp.	Humidity	Wind	Golf Players
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
13	Overcast	Hot	Normal	Weak	44

Golf players for high humidity = {25, 30, 46, 45, 35, 52, 30}

Standard deviation for golf players for high humidity = 9.36

Normal humidity

Golf players for normal humidity = {52, 23, 43, 38, 46, 48, 44}

Standard deviation for golf players for normal humidity = 8.73

Summarizing standard deviations for humidity feature

Humidity	Stdev of Golf Player	Instances
High	9.36	7
Normal	8.73	7

Weighted standard deviation for humidity = (7/14)x9.36 + (7/14)x8.73 = 9.04

Standard deviation reduction for humidity = 9.32 – 9.04 = 0.27

Wind

Wind is a binary class, too. It can either be Strong or Weak.

Strong Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
2	Sunny	Hot	High	Strong	30
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
14	Rain	Mild	High	Strong	30

Golf players for strong wind= {30, 23, 43, 48, 52, 30}

Standard deviation for golf players for strong wind = 10.59

Weak Wind

1	Sunny	Hot	High	Weak	25
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52

8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
13	Overcast	Hot	Normal	Weak	44

Golf players for weakk wind= {25, 46, 45, 52, 35, 38, 46, 44}

Standard deviation for golf players for weak wind = 7.87

Summarizing standard deviations for wind feature

Wind	Stdev of Golf Player	Instances
Strong	10.59	6
Weak	7.87	8

Weighted standard deviation for wind = (6/14)x10.59 + (8/14)x7.87 = 9.03

Standard deviation reduction for wind = 9.32 – 9.03 = 0.29

So, we've calculated standard deviation reduction values for all features. The winner is outlook because it has the highest score.

Feature	Standard Deviation Reduction
Outlook	1.66
Temperature	0.47
Humidity	0.27
Wind	0.29

We'll put outlook decision at the top of decision tree. Let's monitor the new sub data sets for the candidate branches of outlook feature.

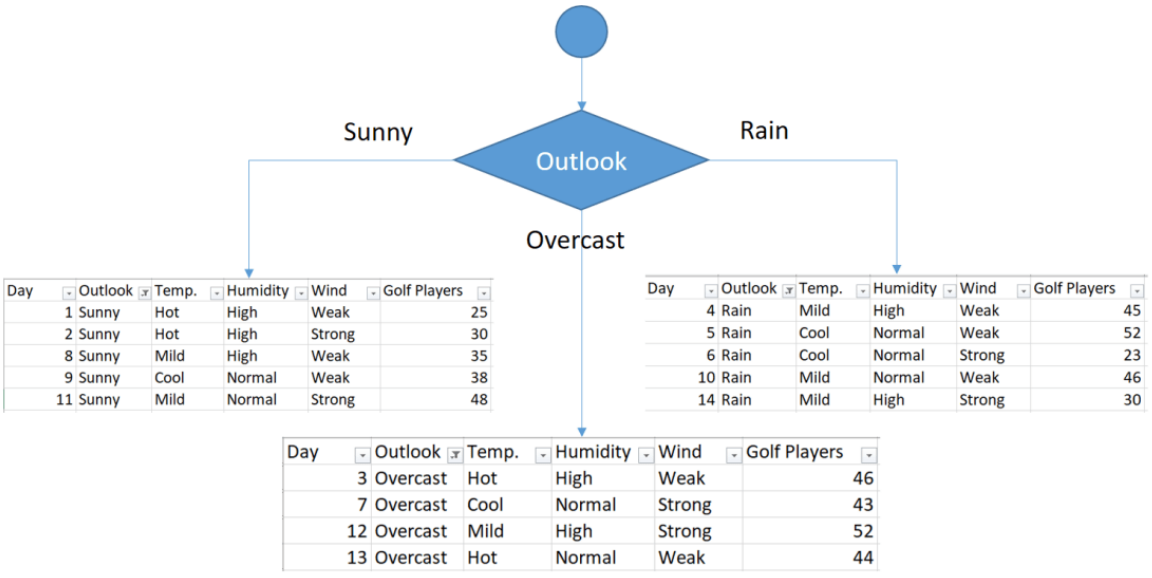


Fig. 1. Putting outlook at the top of the tree

Sunny Outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25

2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Standard deviation for sunny outlook = 7.78

Notice that we will use this standard deviation value as global standard deviation for this sub data set.

Sunny outlook and Hot Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30

Standard deviation for sunny outlook and hot temperature = 2.5

Sunny outlook and Cool Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38

Standard deviation for sunny outlook and cool temperature = 0

Sunny outlook and Mild Temperature

Day	Outlook	Temp.	Humidity	Wind	Golf Players
8	Sunny	Mild	High	Weak	35
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and mild temperature = 6.5

Summary of standard deviations for temperature feature when outlook is sunny

Temperature	Stdev for Golf Players	Instances
Hot	2.5	2
Cool	0	1
Mild	6.5	2

Weighted standard deviation for sunny outlook and temperature = (2/5)x2.5 + (1/5)x0 + (2/5)x6.5 = 3.6

Standard deviation reduction for sunny outlook and temperature = 7.78 – 3.6 = 4.18

Sunny outlook and high humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25

2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35

Standard deviation for sunny outlook and high humidity = 4.08

Sunny outlook and normal humidity

Day	Outlook	Temp.	Humidity	Wind	Golf Players
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and normal humidity = 5

Summarizing standard deviations for humidity feature when outlook is sunny

Humidity	Stdev for Golf Players	Instances
High	4.08	3
Normal	5.00	2

Weighted standard deviations for sunny outlook and humidity = (3/5)x4.08 + (2/5)x5 = 4.45

Standard deviation reduction for sunny outlook and humidity = 7.78 – 4.45 = 3.33

Sunny outlook and Strong Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
2	Sunny	Hot	High	Strong	30
11	Sunny	Mild	Normal	Strong	48

Standard deviation for sunny outlook and strong wind = 9

Sunny outlook and Weak Wind

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38

Standard deviation for sunny outlook and weak wind = 5.56

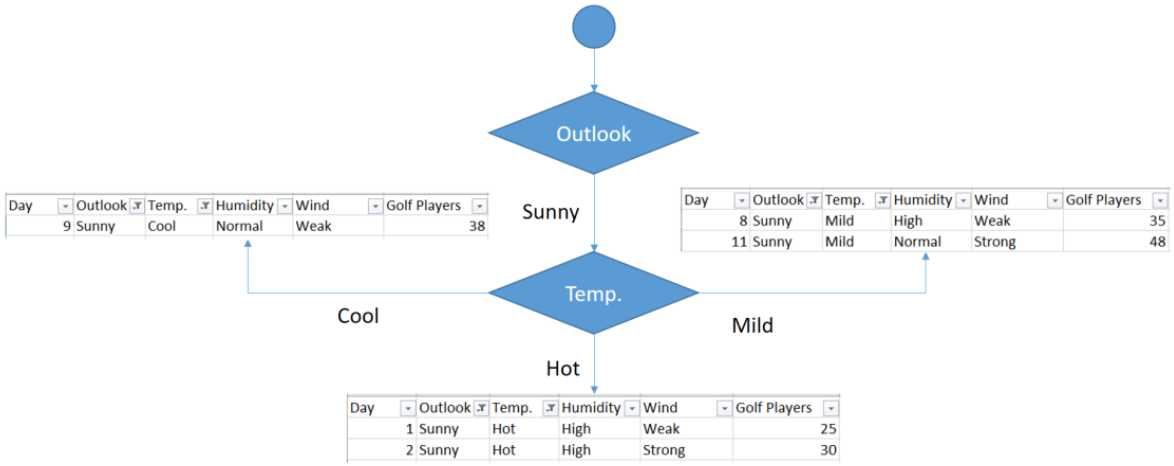
Wind	Stdev for Golf Players	Instances
Strong	9	2
Weak	5.56	3

Weighted standard deviations for sunny outlook and wind = (2/5)x9 + (3/5)x5.56 = 6.93

Standard deviation reduction for sunny outlook and wind = 7.78 – 6.93 = 0.85

We've calculated standard deviation reductions for sunny outlook. The winner is temperature.

Feature	Standard Deviation Reduction
Temperature	4.18
Humidity	3.33
Wind	0.85



Putting temperature decision at the bottom of sunny outlook

Pruning

Cool branch has one instance in its sub data set. We can say that if outlook is sunny and temperature is cool, then there would be 38 golf players. But what about hot branch? There are still 2 instances. Should we add another branch for weak wind and strong wind? No, we should not. Because this causes over-fitting. We should terminate building branches, for example if there are less than five instances in the sub data set. Or standard deviation of the sub data set can be less than 5% of the entire data set. I prefer to apply the first one. I will terminate the branch if there are less than 5 instances in the current sub data set. If this termination condition is satisfied, then I will calculate the average of the sub data set. This operation is called as pruning in decision tree trees.

Overcast outlook

Overcast outlook branch has already 4 instances in the sub data set. We can terminate building branches for this leaf. Final decision will be average of the following table for overcast outlook.

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

If outlook is overcast, then there would be (46+43+52+44)/4 = 46.25 golf players.

Rainy Outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

We need to find standard deviation reduction values for the rest of the features in same way for the sub data set above.

Standard deviation for rainy outlook = 10.87

Notice that we will use this value as global standard deviation for this branch in reduction step.

Rainy outlook and temperature

Temperature	Standard deviation for golf players	instances
Cool	14.50	2
Mild	7.32	3

Weighted standard deviation for rainy outlook and temperature = (2/5)x14.50 + (3/5)x7.32 = 10.19

Standard deviation reduction for rainy outlook and temperature = 10.87 – 10.19 = 0.67

Rainy outlook and humidity

Humidity	Standard deviation for golf players	instances
High	7.50	2
Normal	12.50	3

Weighted standard deviation for rainy outlook and humidity = (2/5)x7.50 + (3/5)x12.50 = 10.50

Standard deviation reduction for rainy outlook and humidity = 10.87 – 10.50 = 0.37

Rainy outlook and wind

Wind	Standard deviation for golf players	instances
Weak	3.09	3
Strong	3.5	2

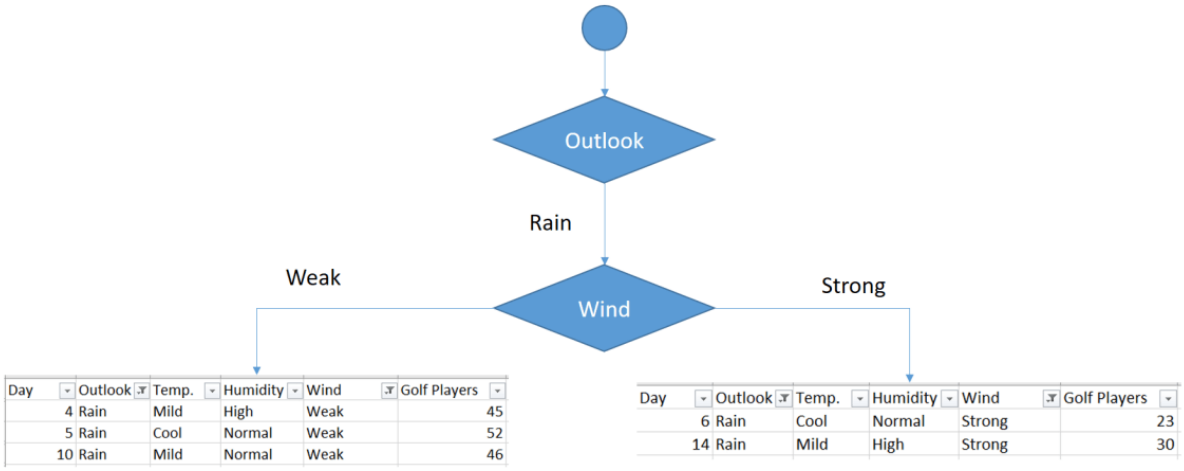
Weighted standard deviation for rainy outlook and wind = (3/5)x3.09 + (2/5)x3.5 = 3.25

Standard deviation reduction for rainy outlook and wind = 10.87 – 3.25 = 7.62

Summarizing rainy outlook

As illustrated below, the winner is wind feature.

Feature	**Standard deviation reduction**
Temperature	0.67
Humidity	0.37
Wind	7.62



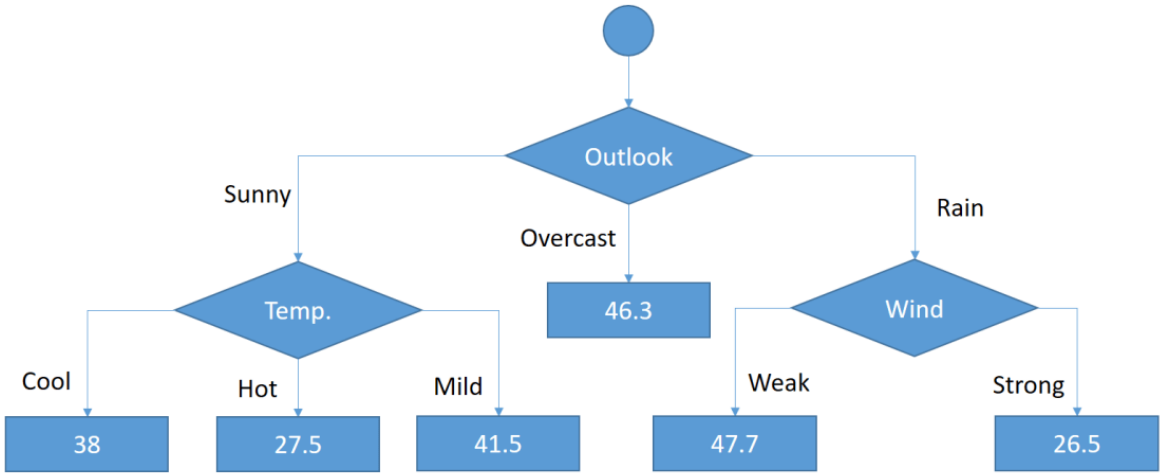
Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
10	Rain	Mild	Normal	Weak	46

Day	Outlook	Temp.	Humidity	Wind	Golf Players
6	Rain	Cool	Normal	Strong	23
14	Rain	Mild	High	Strong	30

Sub data set for rainy outlook

As seen, both branches have items less than 5. Now, we can terminate these leafs based on the termination rule.

So, Final form of the decision tree is demonstrated below.



Final form of the regression tree

Advantages and Disadvantages of Trees Decision trees

1. Trees give a visual schema of the relationship of variables used for classification and hence are more explainable. The hierarchy of the tree provides insight into variable importance.

2. At times they can actually mirror decision making processes.

3. White box model which is explainable and we can track back to each result of the model. This is in contrast to black box models such as neural networks.

4. In general there is less need to prepare and clean data such as normalization and one hot encoding of categorical variables and missing values.

Note the Sklearn implementation currently does not support categorical variables, so we do need to create dummy variables. Similarly it does not support missing values. But both can be handled in theory.

5. Model can be validated statistically

Disadvantages

1. Prone to overfitting and hence lower predictive accuracy

2. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem for example can be mitigated by using decision trees within an ensemble

3. Can be non-robust, i.e., a small change in the data can cause a large change in the final estimated tree

4. Predictions are approximate, based on relevant terminal nodes. Hence it it may not be the best method to extrapolate the results of the model to unseen cases.

5. Decision tree learners create biased trees if some classes dominate. It is required to balance the dataset prior to fitting with the decision tree.