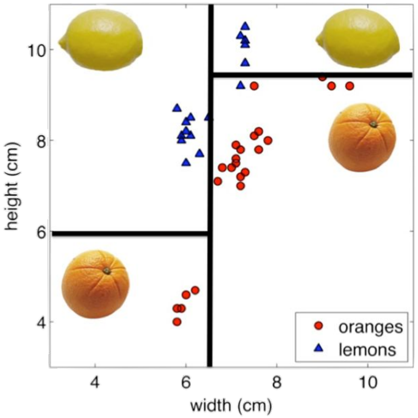
Berkeley Engineering | Berkeley Haas

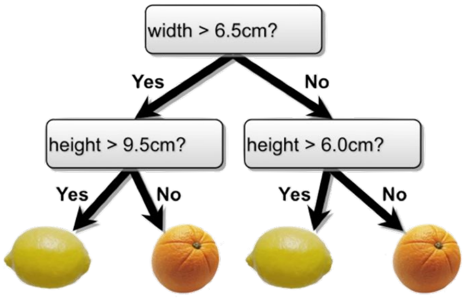
PROFESSIONAL CERTIFICATE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

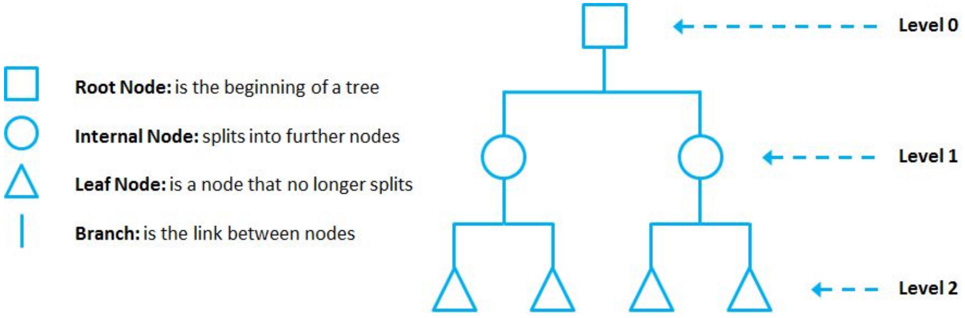
Office Hour #14 with Matilde D'Amelio

Decision Tree

Decision Trees are a type of **Supervised Machine Learning** where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely **decision nodes** and **leaves**. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.





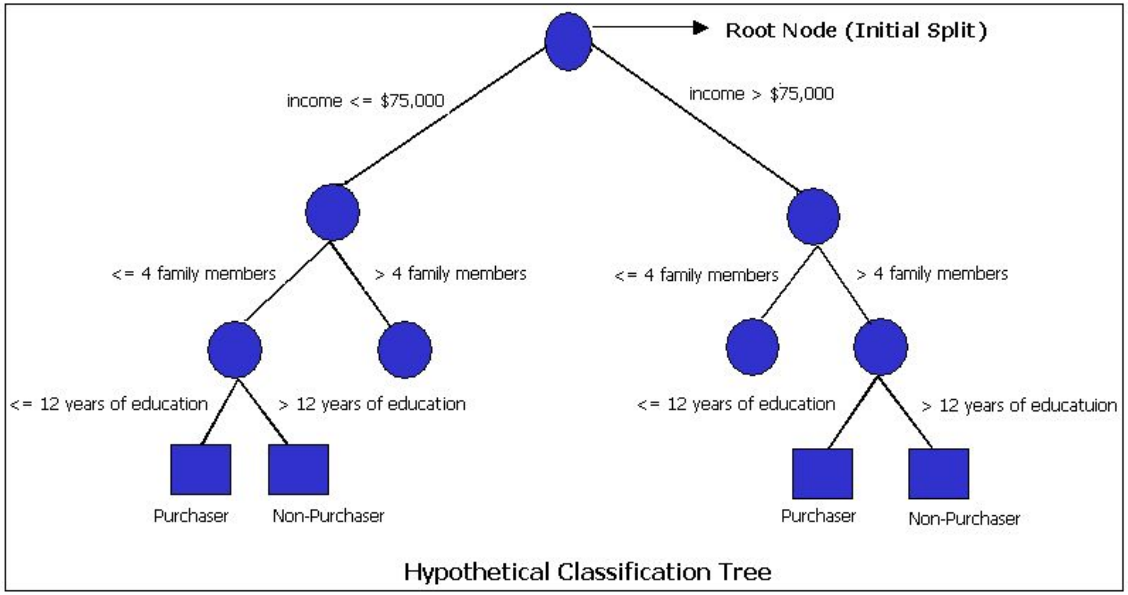

Root Node: is the beginning of a tree

Internal Node: splits into further nodes

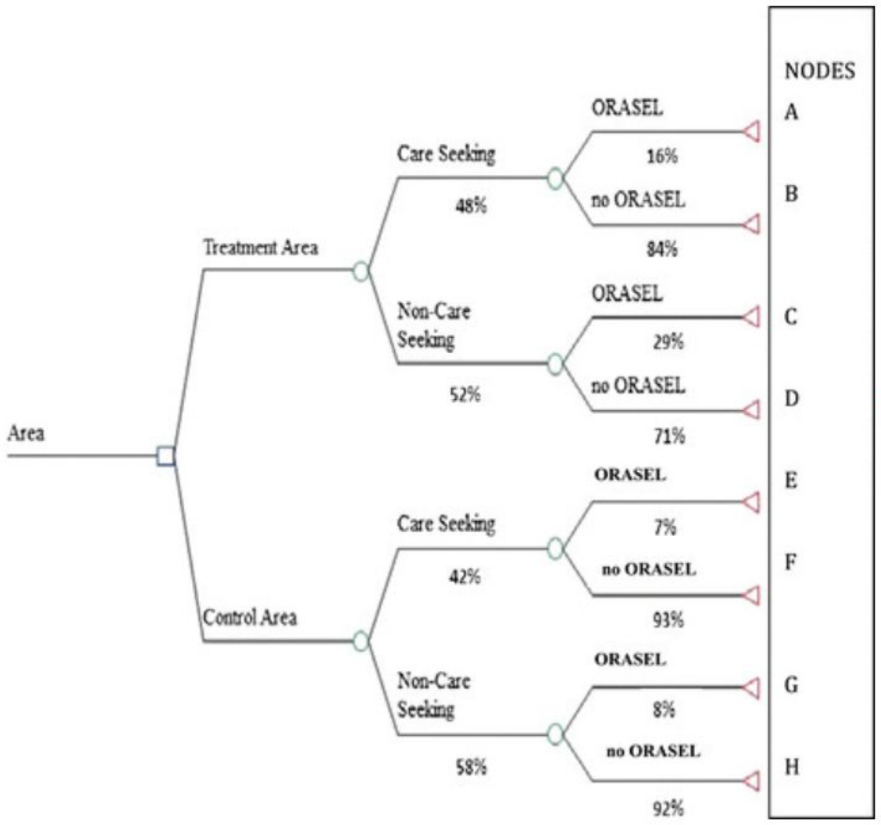
Leaf Node: is a node that no longer splits

Branch: is the link between nodes

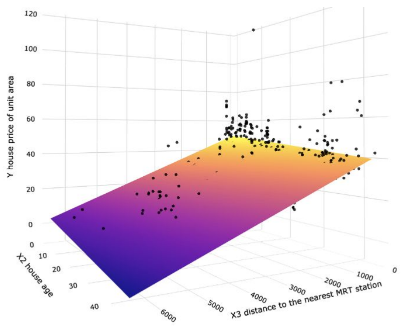
Decision Tree: Classification (Categorical Variables)

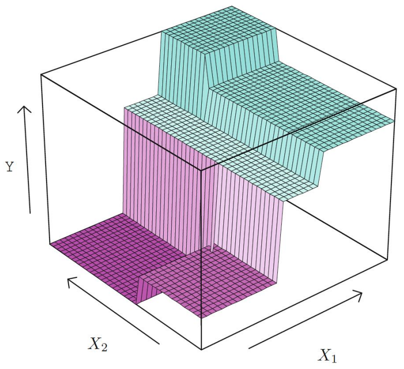


Decision Tree: Regression (Numerical Variables)



This DT examines the cost-effectiveness of achieving increases in the use of oral rehydration solution and zinc supplementation in the management of acute diarrhea in children under 5 years through social franchising.





https://medium.com/@ODSC/the-complete-guide-to-decision-trees-part-1-aa68b34f476d

Decision Tree: Applications

- **Healthcare Industry** : E.g., identify the main risk factors of developing some type of dementia in the future.
- **Chatbot**
- **Environment and Agriculture**: E.g, classify different crop types and identify their phenological stages; identify best way to deal with invasive species
- Perform **sentiment analysi**s of texts, and identify the emotions behind them
- Improve **fraud detection:** e.g., find patterns of transactions and credit cards that match cases of fraud

Decision Tree: Advantages and Disadvantages

ADVANTAGES:

- Simple to understand, interpret, visualize. Trees are very easy to explain to people, and easily interpreted even by a non-expert.
- Decision trees implicitly perform variable screening or feature selection.
- Can handle both numerical and categorical data. Can also handle multi-output problems.
- Decision trees require relatively little effort from users for data preparation.
- Nonlinear relationships between parameters do not affect tree performance.
- Some people believe that decision trees more closely mirror human decision-making.

DISADVANTAGES:

- Over-complex trees that do not generalize the data well: overfitting.
- Small variations in the data might result in a completely different tree: variance.
- Decision tree learners create biased trees if some classes dominate.
- Sensitive to changes in the training data

Steps to Build a Decision Tree

- Chose the approach to divide branches (Divide & Conquer vs Greedy Principles)
- Choose how create a split
  - Categorical (child node per category, per sub-category or binary)
  - Numerical (binary - average)
- Define criteria to assess the quality of the split
  - Categorical
    - Define the threshold of Information gain (IG)
    - Assess IG with Gini Index or Entropy
  - Numerical
    - MSE
- Define when you do stop splitting (pre-pruning, post-pruning, random forest)

Decision Tree: Splitting Choices

While there is no ‘correct’ way to define an optimal split, there are some common sensical guidelines for every splitting criterion:

- the regions in the feature space should **grow progressively more pure with the number of splits**. That is, we should see each region ‘specialize’ towards a single class.
- we shouldn’t end up with **empty regions** - regions containing no training points.

Entropy

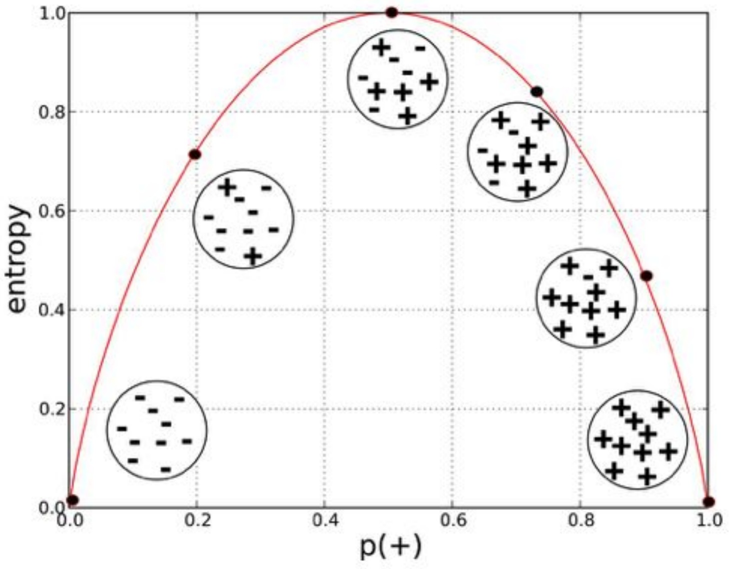
Higher entropy means the distribution is uniform-like (flat histogram) and thus values sampled from it are ‘less predictable’ (all possible values are equally probable).

Lower entropy means the distribution has more defined peaks and valleys and thus values sampled from it are ‘more predictable’ (values around the peaks are more probable).

It is the measure of disorder or heterogeneity of my dataset

Decision Tree: Entropy

$$Entropy=\sum_{i=1}^{C}-p_i*\log_2(p_i)$$



So if we had a total of 100 data points in our dataset with 30 belonging to the positive class and 70 belonging to the negative

$$-\frac{3}{10}\times\log_2\left(\frac{3}{10}\right)-\frac{7}{10}\times\log_2\left(\frac{7}{10}\right)\approx 0.88$$

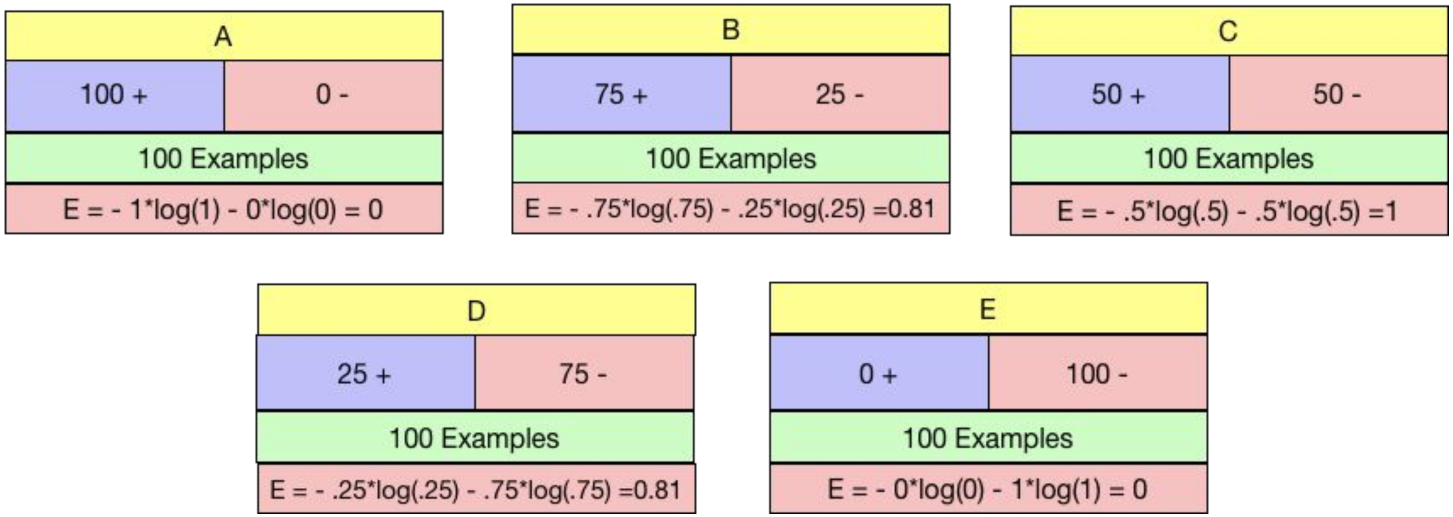
If the sample is completely homogeneous the entropy is zero and if the sample is equally divided then it has entropy of one.

Information Gain: The information gain is based on the decrease in entropy after a data-set is split on an attribute.

Information Gain = how much Entropy we removed

https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Decision Tree: Entropy



A

100 + | 0 -

100 Examples

E = - 1*log(1) - 0*log(0) = 0

B

75 + | 25 -

100 Examples

E = - .75*log(.75) - .25*log(.25) =0.81

C

50 + | 50 -

100 Examples

E = - .5*log(.5) - .5*log(.5) =1

D

25 + | 75 -

100 Examples

E = - .25*log(.25) - .75*log(.75) =0.81

E

0 + | 100 -

100 Examples

E = - 0*log(0) - 1*log(1) = 0

https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

Entropy

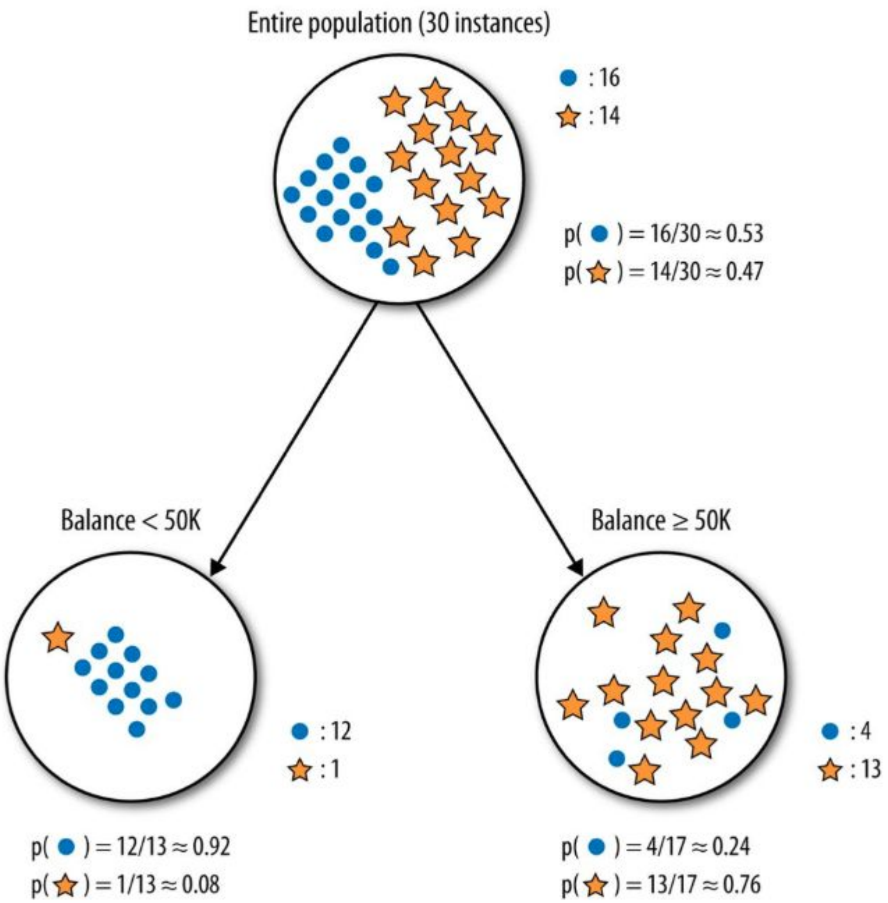
Consider an example where we are building a decision tree to predict whether a loan given to a person would result in a write-off or not. Our entire population consists of 30 instances. 16 belong to the write-off class and the other 14 belong to the non-write-off class. We have two features, namely “**Balance**” that can take on two values -> “< 50K” or “>50K” and “**Residence**” that can take on three values -> “OWN”, “RENT” or “OTHER”.

How a decision tree algorithm would decide what attribute to split on first and what feature provides more information, or reduces more uncertainty about our target variable?

The dots are the data points with class right-off and the stars are the non-write-offs

The entropy for the parent node : 0.62

Feature 1: Balance



Entropy

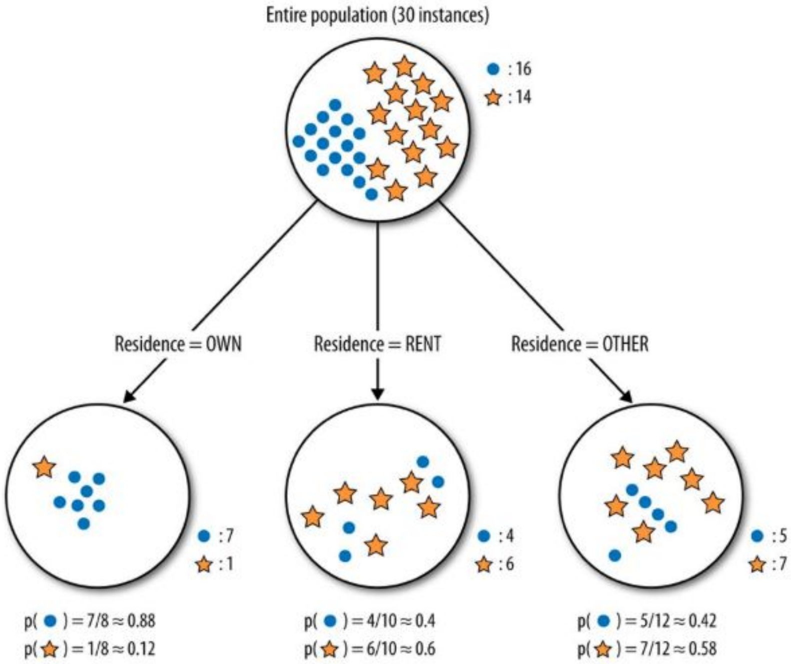
Splitting the tree on Residence gives us 3 child nodes.

The entropy for the parent node : 0.86

The child nodes from splitting on Balance do seem purer than those of Residence. However the left most node for residence is also very pure but this is where the weighted averages come in play. Even though that node is very pure, it has the least amount of the total observations and a result contributes a small portion of it's purity when we calculate the total entropy from splitting on Residence.

By itself the feature, Balance provides more information about our target variable than Residence. It reduces more disorder in our target variable. A **decision tree algorithm would use this result to make the first split on our data using Balance. From here on, the decision tree algorithm would use this process at every split to decide what feature it is going to split on next**.

Feature 2: Residence

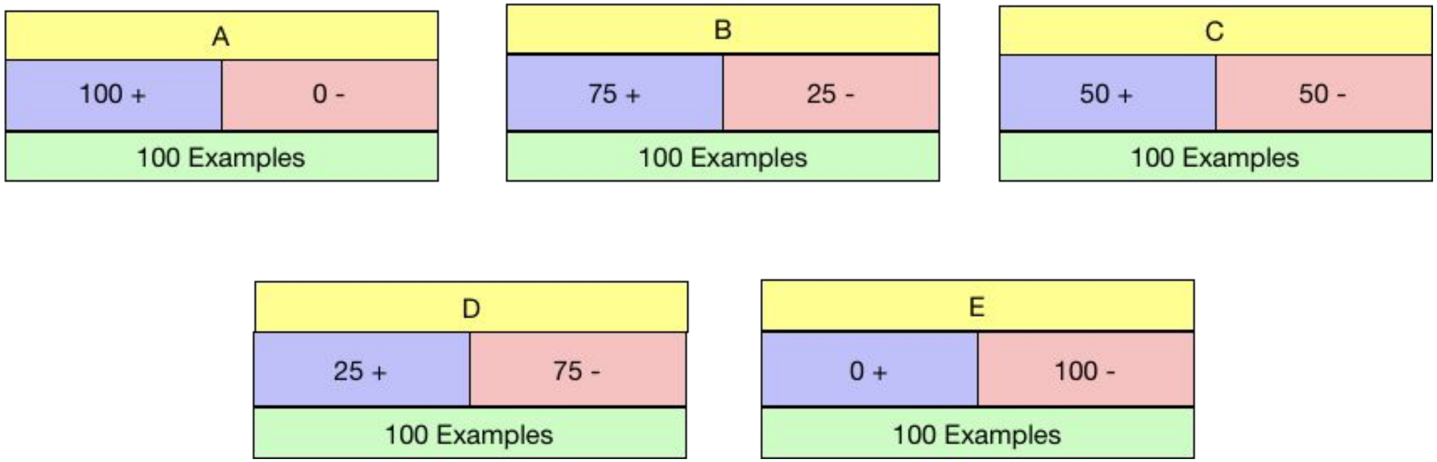


Gini Index

$$I_G(n) = 1 - \sum_{i=1}^{J} (p_i)^2$$

In simple terms, it is the measure of impurity in a node expressed in probability terms

E.g. five examples of candidate nodes, which is the ideal situation to be in?



A	
100 +	0 -
100 Examples	

B	
75 +	25 -
100 Examples	

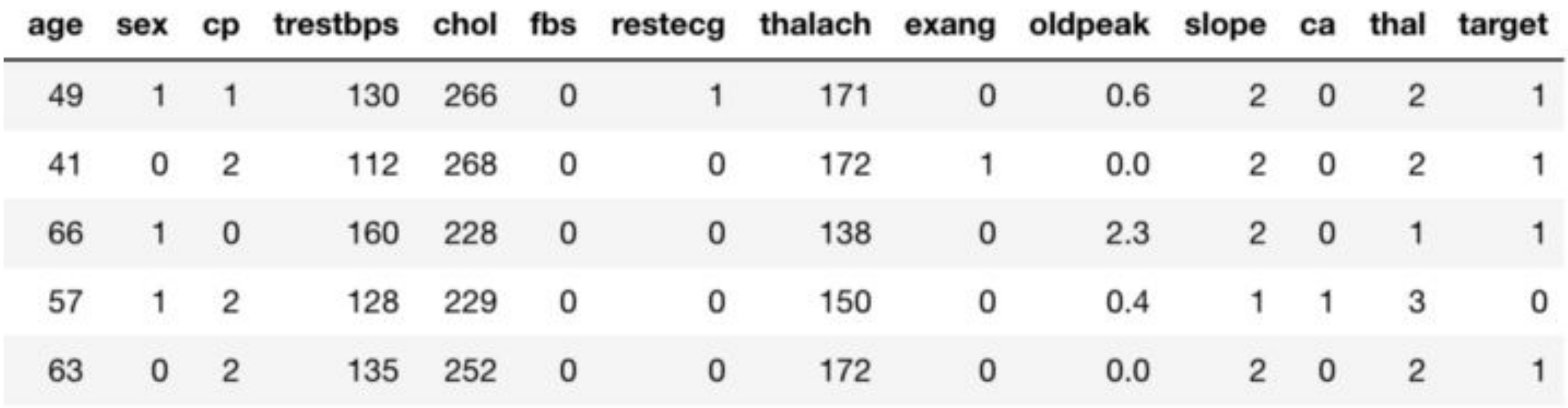
C	
50 +	50 -
100 Examples	

D	
25 +	75 -
100 Examples	

E	
0 +	100 -
100 Examples	

https://towardsdatascience.com/the-simple-math-behind-3-decision-tree-splitting-criterions-85d4de2a75fe

Gini Index

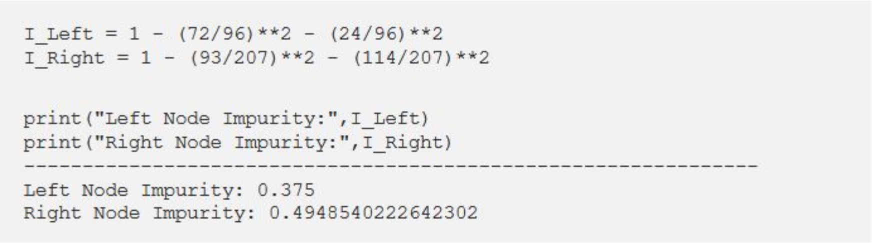


age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
41	0	2	112	268	0	0	172	1	0.0	2	0	2	1
66	1	0	160	228	0	0	138	0	2.3	2	0	1	1
57	1	2	128	229	0	0	150	0	0.4	1	1	3	0
63	0	2	135	252	0	0	172	0	0.0	2	0	2	1

UCI Heart Disease data. The “target” field refers to the presence of heart disease in the patient. It is 0 (no presence) or 1

https://towardsdatascience.com/the-simple-math-behind-3-decision-tree-splitting-criterions-85d4de2a75fe

Gini Index

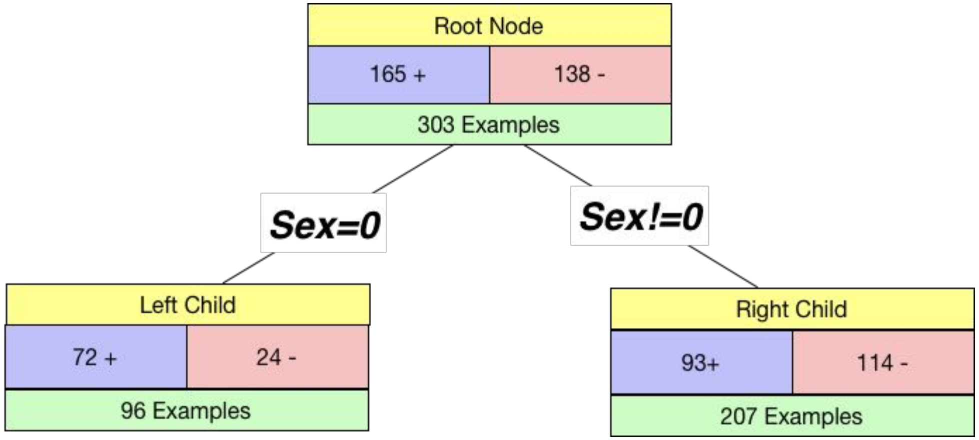


```
I_Left = 1 - (72/96)**2 - (24/96)**2
I_Right = 1 - (93/207)**2 - (114/207)**2


print("Left Node Impurity:",I_Left)
print("Right Node Impurity:",I_Right)
-----------------------------------------------------------------
Left Node Impurity: 0.375
Right Node Impurity: 0.4948540222642302
```

Categorical variable split (e.g. Sex)

we take a weighted average of the two impurities weighted by the number of examples in the individual node



Root Node

165 + | 138 -

303 Examples

Sex=0

Sex!=0

Left Child

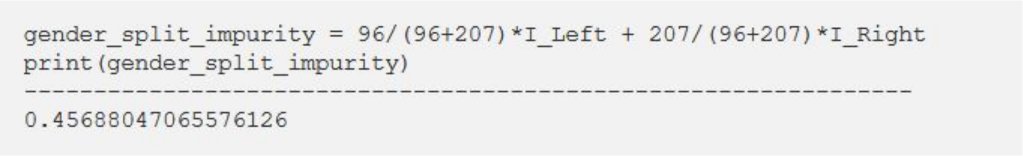
72 + | 24 -

96 Examples

Right Child

93+ | 114 -

207 Examples



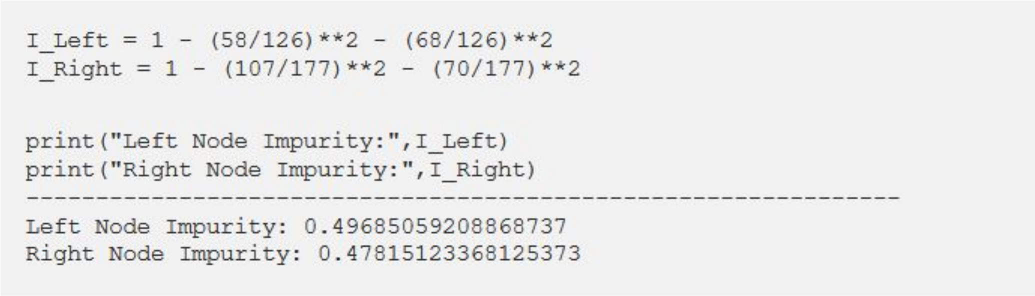
```
gender_split_impurity = 96/(96+207)*I_Left + 207/(96+207)*I_Right
print(gender_split_impurity)
----------------------------------------------------------------
0.45688047065576126
```

https://towardsdatascience.com/the-simple-math-behind-3-decision-tree-splitting-criterions-85d4de2a75fe

Gini Index

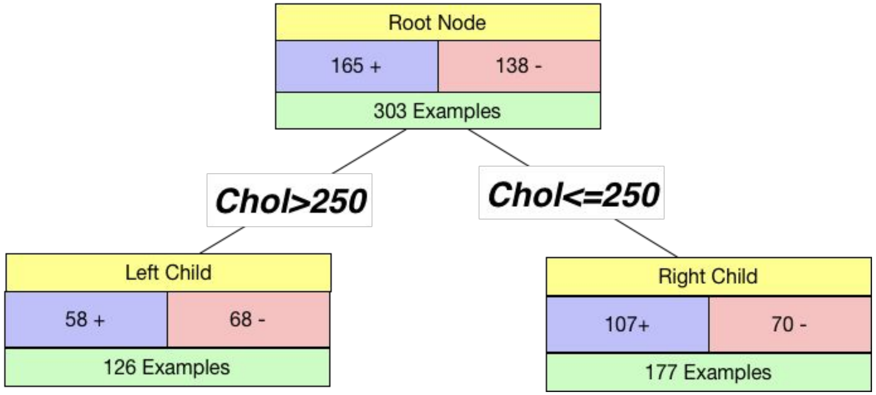
Continuous variable split

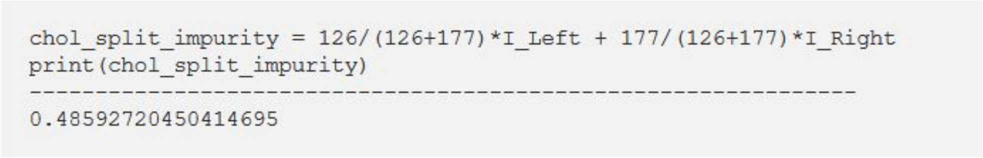
(e.g. Cholesterol)



```
I_Left = 1 - (58/126)**2 - (68/126)**2
I_Right = 1 - (107/177)**2 - (70/177)**2

print("Left Node Impurity:",I_Left)
print("Right Node Impurity:",I_Right)
----------------------------------------------------------------
Left Node Impurity: 0.49685059208868737
Right Node Impurity: 0.47815123368125373
```

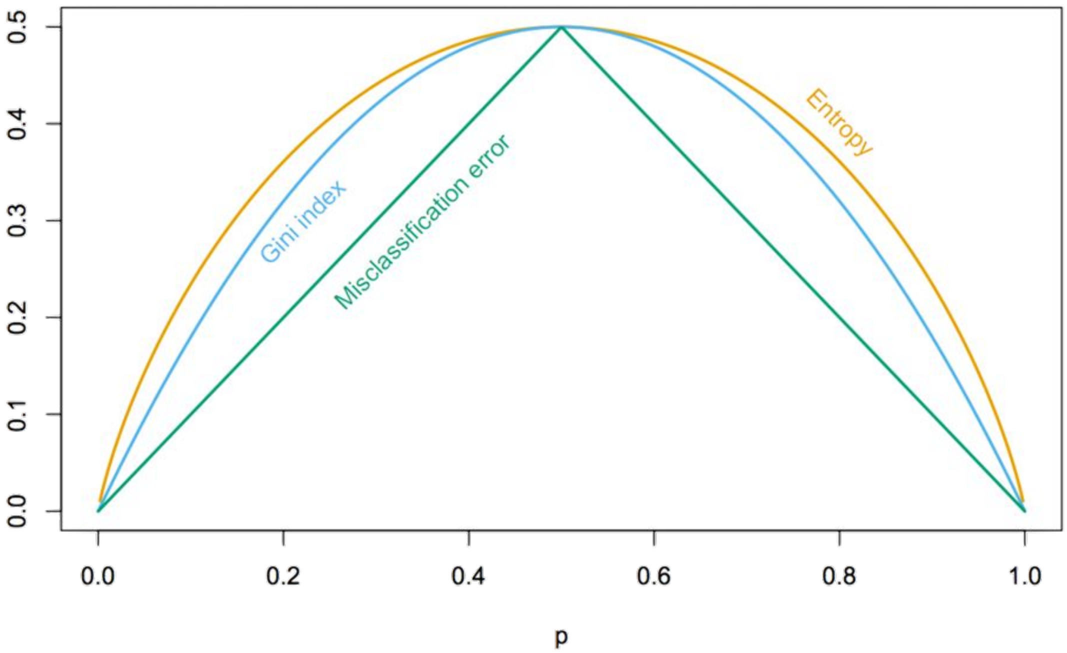




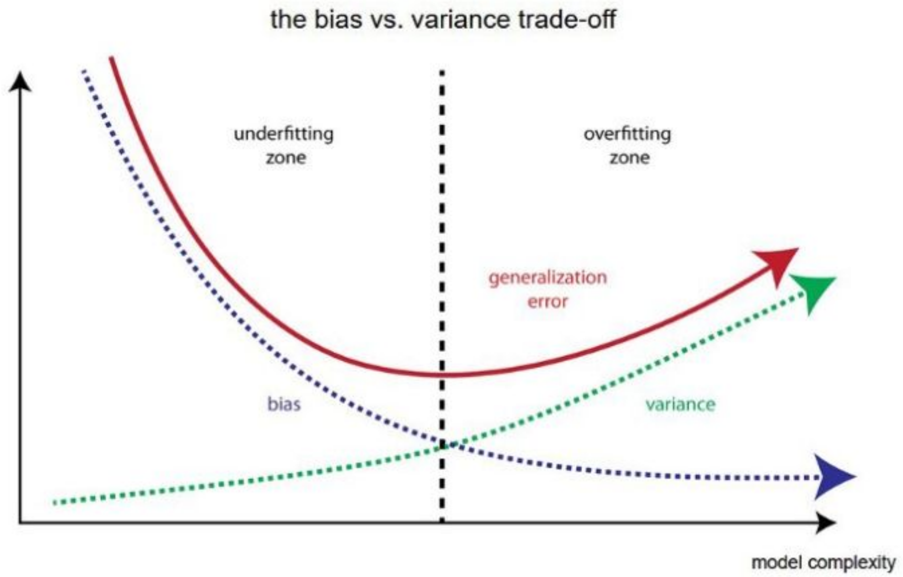
```
chol_split_impurity = 126/(126+177)*I_Left + 177/(126+177)*I_Right
print(chol_split_impurity)
-------------------------------------------------------------
0.48592720450414695
```

https://towardsdatascience.com/the-simple-math-behind-3-decision-tree-splitting-criterions-85d4de2a75fe

Comparison of Splitting Criteria



Stopping Criteria: Bias-Variance trade off



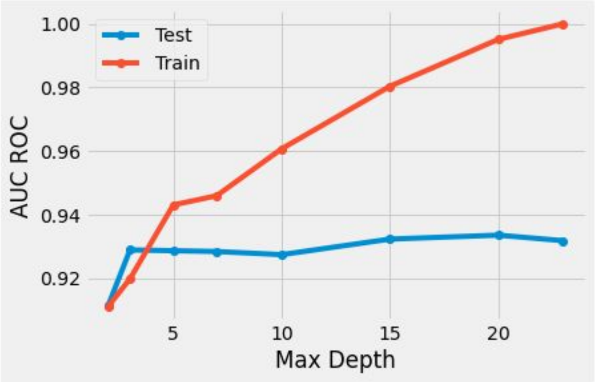
Techniques:

- Pre-pruning
- Post-pruning
- Random Forest

Pre pruning

The pre-pruning technique refers to the early stopping of the growth of the decision tree. The pre-pruning technique involves tuning the hyperparameters of the decision tree model prior to the training pipeline. The hyperparameters of the decision tree including **max_depth, min_samples_leaf, min_samples_split** can be tuned to early stop the growth of the tree and prevent the model from overfitting.

As observed from the plot, with an increase in max_depth training AUC-ROC score continuously increases, but the test AUC score remains constants after a value of max depth. The best-fit decision tree is at a max depth value of 5. Increase the max depth value further can cause an overfitting problem. max_depth, min_samples_leaf, min_samples_splitare other hyperparameters of the decision tree algorithm that can be tuned to get a robust model.

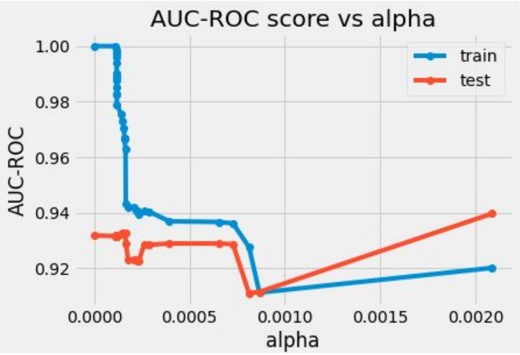


(Image by Author), AUC-ROC score vs max depth

Post-Pruning

The Post-pruning technique allows the decision tree model to grow to its full depth, then removes the tree branches to prevent the model from overfitting.

Cost complexity pruning (ccp) is one type of post-pruning technique. In case of cost complexity pruning, the ccp_alpha can be tuned to get the best fit model.



- Train decision tree classifiers with different values of ccp_alphas and compute train and test performance scores.
- Plot train and test scores for each value of ccp_alphas values.

From the above plot, ccp_alpha=0.000179 can be considered as the best parameter as AUC-ROC scores for train and test are 0.94 and 0.92 respectively.

Group Discussion

What can be applications of the Decision Tree in your Capstone Project?

.

QUESTIONS?