Introduction to the basics of AI

Session 6

Z. TAIA-ALAOUI

Outline

- Definitions
- Decision Trees / Random Forest
- Implementation

Statistical Tools - Dataset

- **Set of N samples expressed in a space of p Variables**

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{N1} & \cdot & \cdot & \cdot & x_{Np} \end{pmatrix} = [V_1, V_2, \ldots, V_p] = \begin{bmatrix} X_1^{\mathrm{T}} \\ X_2^{\mathrm{T}} \\ \cdot \\ \cdot \\ \cdot \\ X_N^{\mathrm{T}} \end{bmatrix}$$

V_j (column) and X_i (row)

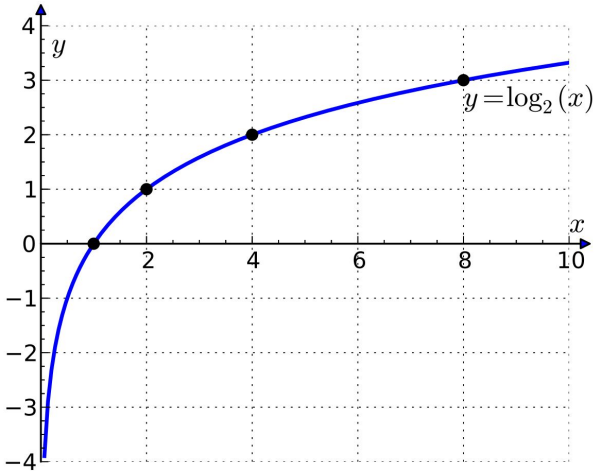
$$X_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \cdot \\ \cdot \\ \cdot \\ x_{ip} \end{bmatrix} \qquad V_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \cdot \\ \cdot \\ \cdot \\ x_{Nj} \end{bmatrix}$$

Definitions

- **Information Theory**

- **Low Probability Event**: High Information (*surprising*) → words like verbs and adjectives in a sentence.
- **High Probability Event**: Low Information (*unsurprising*) → words like “and”, “or”, “I” in a sentence.
- Rare events need more information to represent them

```
Information(x) = -ln(p(x))
```

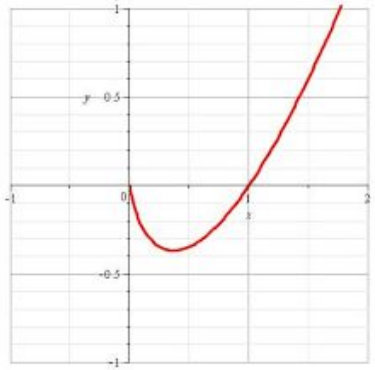


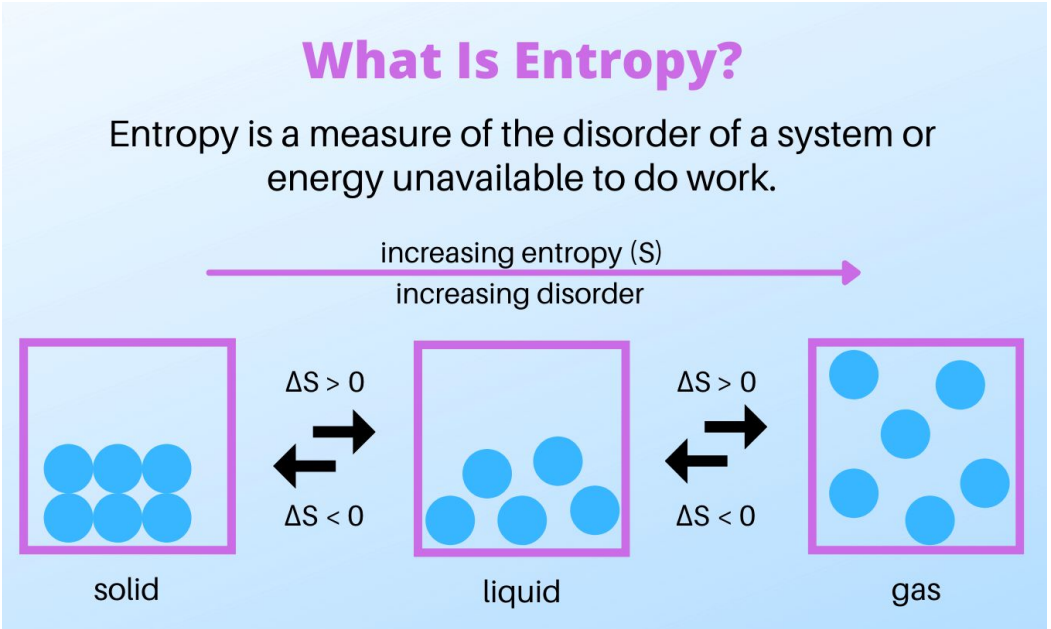
Definitions

- **Information Theory**
- Calculating the information for a random variable is called “information entropy,” “Shannon entropy,” or simply “entropy“.

$$p\colon \mathcal{X} \to [0, 1]$$

$$\mathrm{H}(X) := -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$



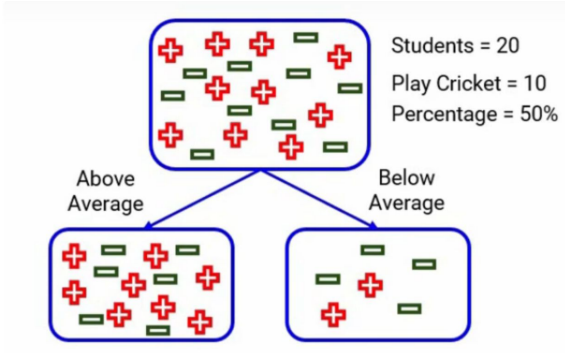


Definitions

- Gini impurity (1 - Gini)

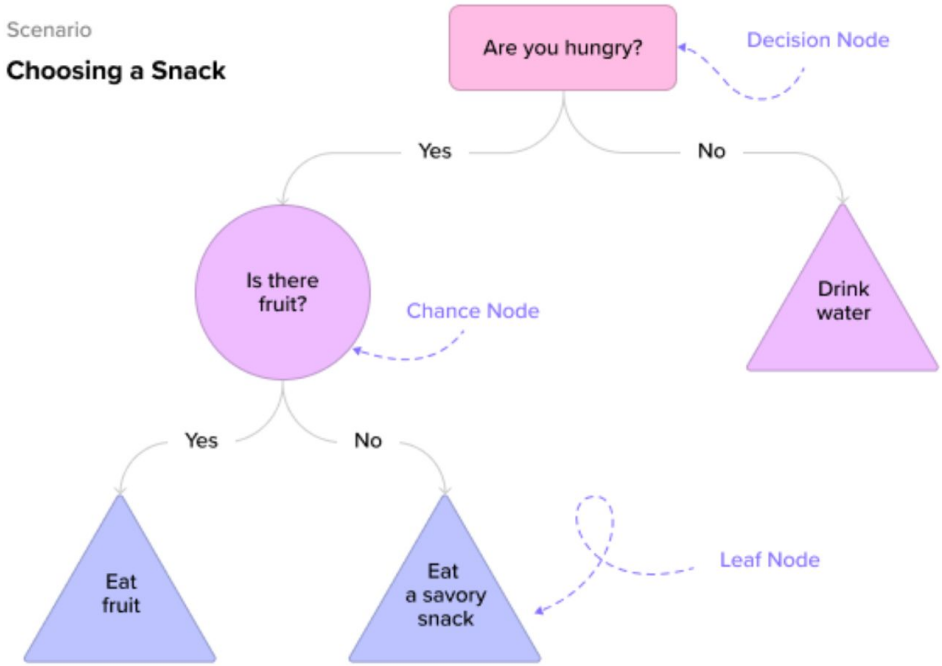
Consider a dataset D that contains samples from k classes. The probability of samples belonging to class i at a given node can be denoted as p_i. Then the Gini Impurity of D is defined as:

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$



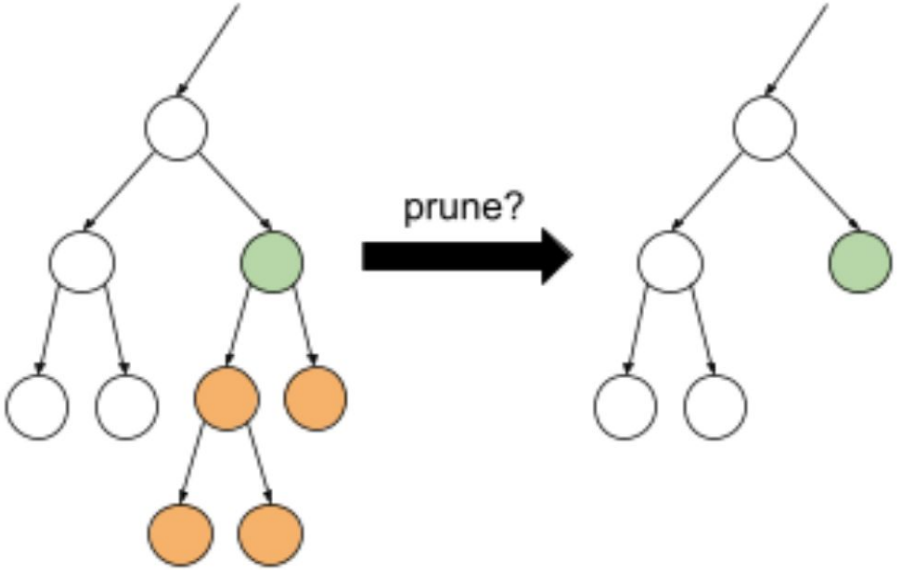
Definitions

- **Root node**: The topmost node of a decision tree that represents the entire message or decision
- **Decision (or internal) node**: A node within a decision tree where the prior node branches into two or more variables
- **Leaf (or terminal) node**: The leaf node is also called the external node or terminal node, which means it has no child—it's the last node in the decision tree and furthest from the root node



Definitions

- **Splitting**: The process of dividing a node into two or more nodes. It's the part at which the decision branches off into variables
- **Pruning**: The opposite of splitting, the process of going through and reducing the tree to only the most important nodes or outcomes



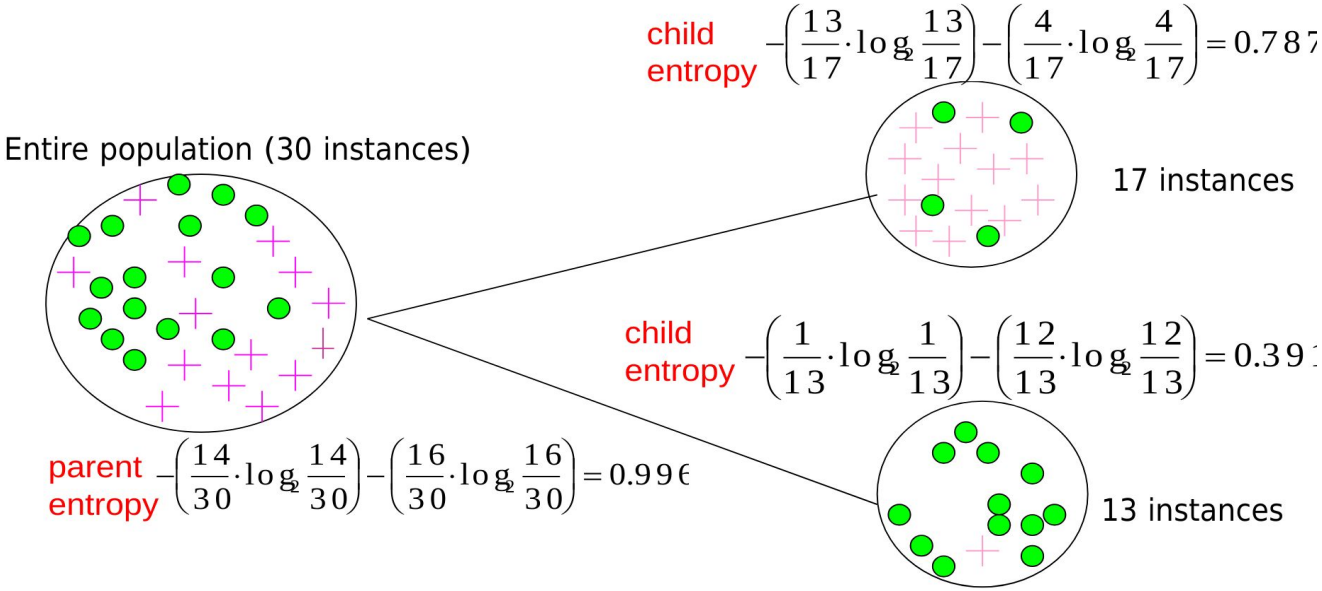
Definitions

- **Information Gain**
  - Information Gain = entropy(parent) – [average entropy(children)]
  - We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.
  - Information gain tells us how important a given attribute of the feature vectors is.
  - It is used it to decide the ordering of attributes in the nodes of a decision tree.

Definitions

Calculating Information Gain

Information Gain = entropy(parent) – [average entropy(children)]



(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38 for this split

ID3 Algorithm

What are the steps in ID3 algorithm?

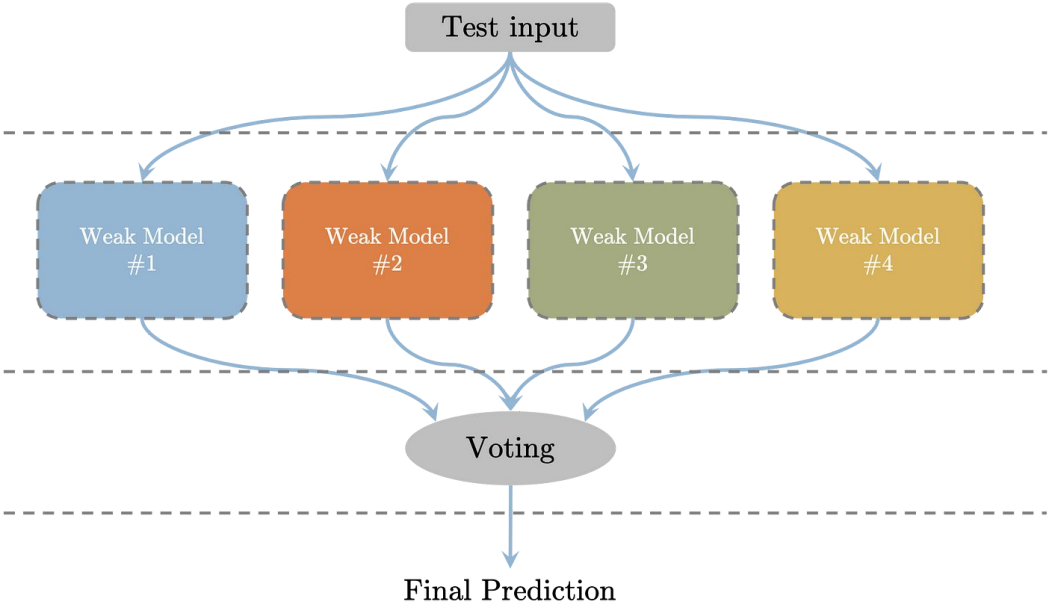
1. **Determine entropy** for the overall the dataset using class distribution.
2. For each feature.
    - Calculate **Entropy for Categorical Values**.
    - Assess **information gain** for each unique categorical value of the feature.
3. Choose the feature that generates **highest information gain**.
4. Iteratively apply all above steps to build the decision tree structure.

ID3 Algorithm

1. ID3 can overfit the training data (to avoid overfitting, smaller decision trees should be preferred over larger ones).
2. This algorithm usually produces small trees, but it does not always produce the smallest possible tree.
3. ID3 is harder to use on continuous data (if the values of any given attribute is continuous, then there are many more places to split the data on this attribute, and searching for the best value to split by can be time-consuming).

Ensemble Learning

Ensemble learning is a machine learning technique that enhances accuracy and resilience in forecasting by merging predictions from multiple models. It aims to mitigate errors or biases that may exist in individual models by leveraging the collective intelligence of the ensemble.



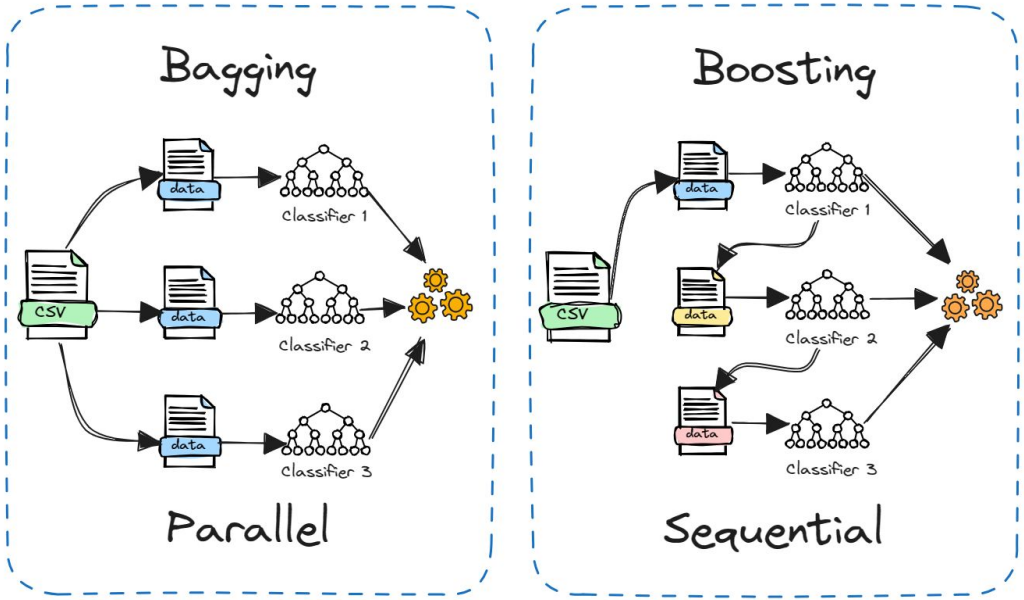
Ensemble Learning

- **Bagging**:

  Bagging (bootstrap aggregating) is an ensemble method that involves training multiple models independently on random subsets of the data, and aggregating their predictions through voting or averaging.

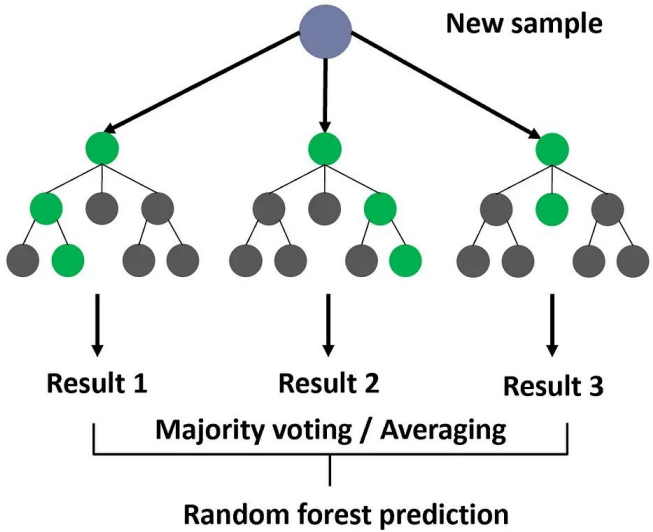
- **Boosting**:

  In boosting, models are trained sequentially, with each model learning from the errors of the previous one. Additionally, bagging typically involves simple averaging of models, while boosting assigns weights based on accuracy.



Random Forest

- **Step 1**: In the Random forest model, a subset of data points and a subset of features is selected for constructing each decision tree. Simply put, n random records and m features are taken from the data set having k number of records.
- **Step 2**: Individual decision trees are constructed for each sample.
- **Step 3**: Each decision tree will generate an output.
- **Step 4**: Final output is considered based on *Majority Voting or Averaging* for Classification and regression, respectively.



Random Forest

- **Diversity**: Not all attributes/variables/features are considered while making an individual tree; each tree is different.
- Immune to the curse of dimensionality: Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization**: Each tree is created independently out of different data and attributes. This means we can fully use the CPU to build random forests.
- **Train-Test split**: In a random forest, we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability**: Stability arises because the result is based on majority voting/ averaging.

Sources

https://en.wikipedia.org/wiki/Entropy_(information_theory)

https://machinelearningmastery.com/what-is-information-entropy/

https://github.com/akshay-madar/decisionTree-from-scratch/blob/master/Decision%20Tree%20Classification%20from%20scratch.ipynb

https://slickplan.com/blog/what-is-a-decision-tree#:~:text=There%20are%20two%20types%20of,branches%20to%20make%20them%20whole

https://sciencenotes.org/wp-content/uploads/2021/11/What-Is-Entropy-Definition.png

https://www.learndatasci.com/glossary/gini-impurity/

https://homes.cs.washington.edu/~shapiro/EE596/notes/InfoGain.pdf

https://www.datacamp.com/tutorial/what-bagging-in-machine-learning-a-guide-with-examples

https://www.kaggle.com/code/tcvieira/simple-random-forest-iris-dataset

https://miro.medium.com/v2/resize:fit:1400/format:webp/1*1woLesOTZE0t3EswiT0mnA.png