Classification : Basic Concepts and Techniques (1)

I_ Introduction to Classification

→ What is Classification

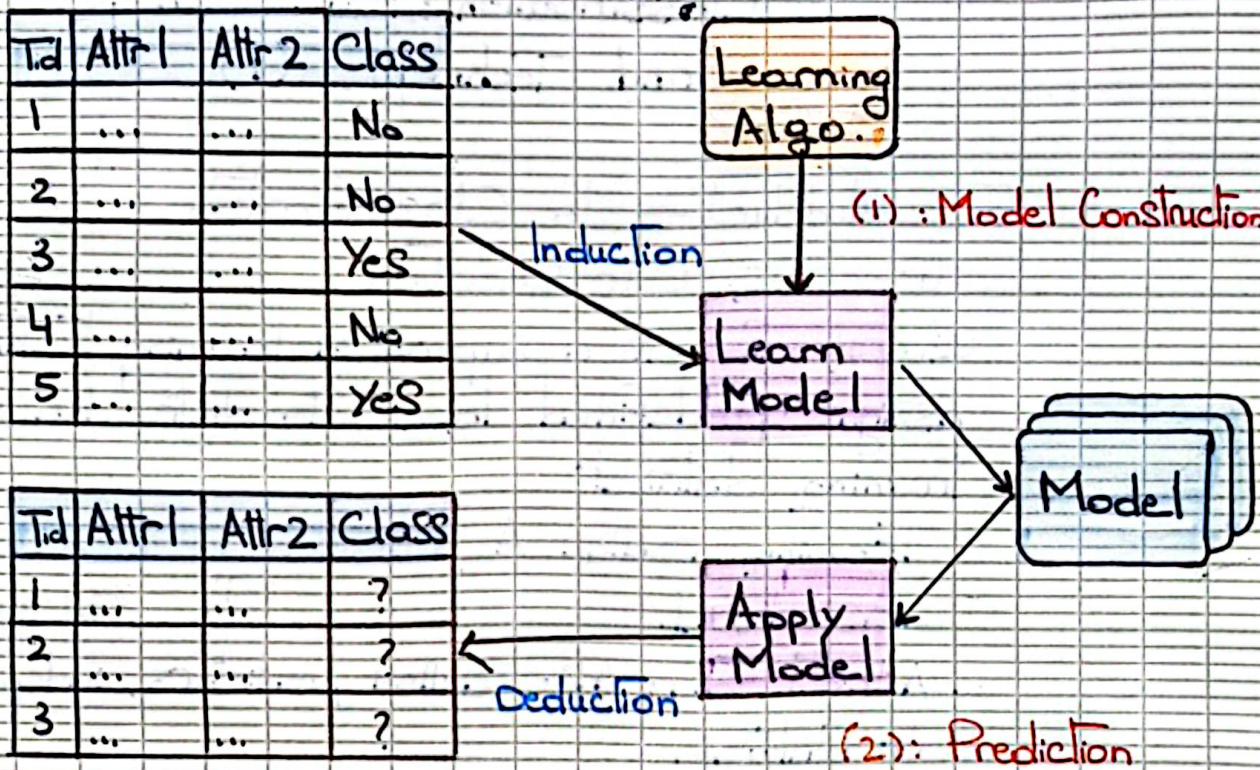
Classification is a supervised learning technique that assigns predefined labels to data based on a training dataset

- Training Data: A collection of records with Known class label.
- Attribute (x): Also called predictors, independ variables or features.
- Class Labels (y): Also called responses, dependent variables, or outputs.

→ General Approach for Classification

1. Model Construction
   - Learn a classification model from the training dataset
   - Output: Decision rules, decision trees, or mathematical models
   - Example:
     Rule: if age > 30 and income > $50K, then "approved loan"

2. Model Prediction
   - Use the model to classify new data points (test)
   - Measure accuracy : Acc = Correct pred / Total pred



II. Classification Techniques

↳ Overview

Classification Techniques are broadly divided into Two categories :

1) Base Classifiers : Individual algorithms for classi.
2) Ensemble Classifiers: Combine multiple base classifiers to improve accuracy

↳ Base Classifiers

1. Decision Tree - Based Methods

- Description: Models data as a tree structure with nodes representing decisions and branches indicating outcomes
- Key Algorithm Examples : ID3, C4.5, CART

. Example:
↳ Decision: "Is income > $30K"
↳ Outcomes: "Loan Approved" or "Loan Rejected"

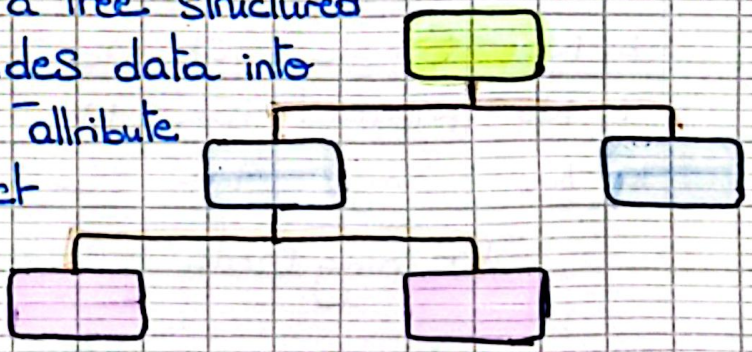
2. Rule - base Methods
3. Nearest _neighbor
4. Neural Networks . Deep Neural Nets
5. Naive Bayes
6. Support Vector Machines (SVM)

↳ Ensemble Classifiers
1. Boosting
2. Bagging (Bootstrap Aggregation)

III. Decision Trees and their Splitting Criteria

↳ What are Decision Trees?

. A decision tree is a tree structured classifier that divides data into subsets based on attribute values, helping predict outcomes



. Nodes:
↳ Internal / split nodes represent attributes
↳ Branches represent decision rules
↳ Decision / Leaf nodes represent class labels
↳ Root Node: Starting point of the tree

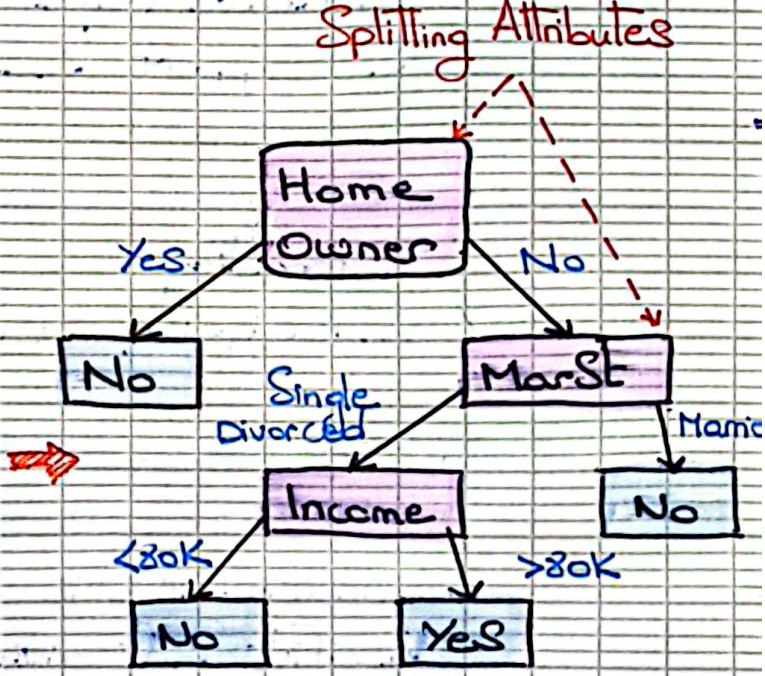
Process:

1) Start from the root.
2) Split data based on attributes
3) Continue until leaf nodes contain homogeneous classes or meet stopping criteria.

Example of a Decision Tree

	cat.	cat.	cont.	class
ID	Home Owner	Marital Status	Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



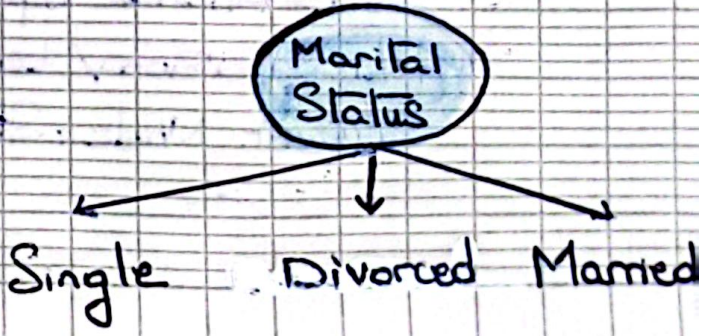
Model: Decision Tree

There could be more than one tree that fits the same data.

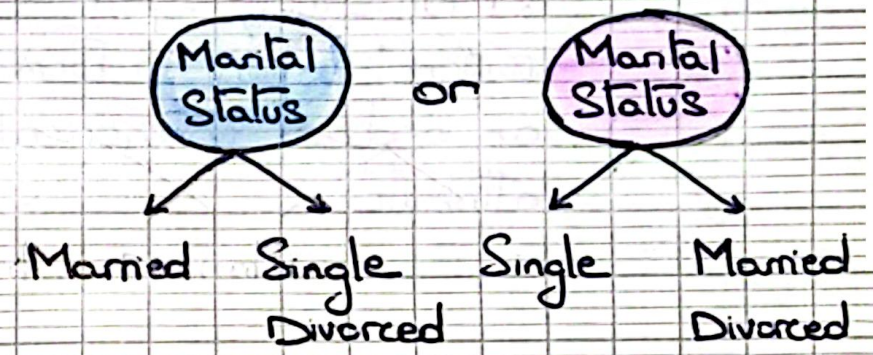
Splitting Criteria

1. Nominal Attributes

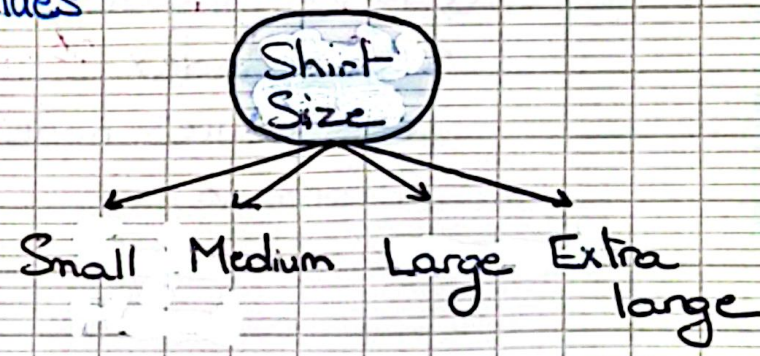
- Multi-way Split: Create a branch for each distinct value



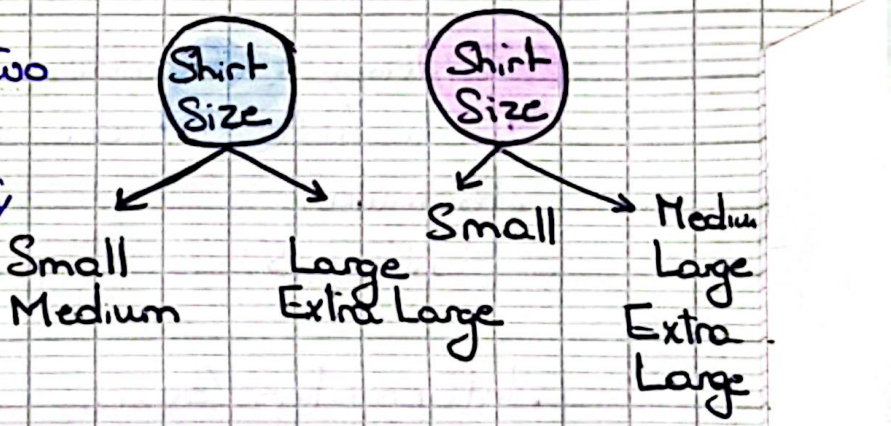
↳ Binary Split
Divides values into two subsets



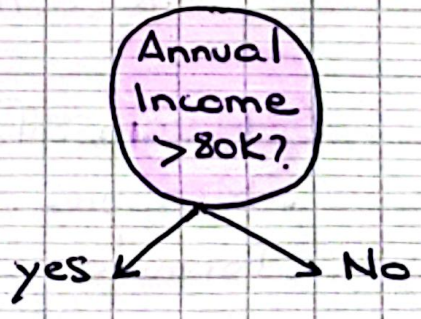
2. Ordinal Attributes
Maintains order among values
↳ Multi-way Split
Use as many partition as distinct values



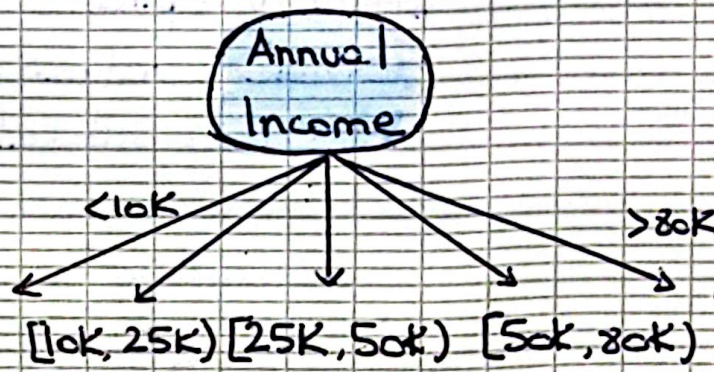
↳ Binary Split
. Divides values into two subsets
. Preserve order property among attribute values



3. Continuous Attributes
Typically requires identifying
↳ Binary Split (A<v) or (A≥v)
. Consider all posible splits and finds the best cut
. Can be more compute intensive.



↳ Multi-way Split
Discretization to form an ordinal categorical attribute



↳ Node Impurity and Splitting Criteria

1. Gini Index
- Measures node impurity based on class probabilities
- Formula:

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

p_i → Proportion of instances in class i

2. Entropy (Information Gain)
- Measures disorder in a dataset
- Formula:

$$Entropy(S) = -\sum_{i=1} p_i \log_2(p_i)$$

- Information Gain:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} |S_v|/|S| \cdot Entropy(S_v)$$

3. Misclassification Error
- Measures the proportion of misclassified records
- Formula:

$$Error(t) = 1 - \max(p_i)$$

Finding the Best Split

1. Compute impurity (e.g. Entropy, Gini Index) before splitting (B)
2. Compute impurity after splitting (A)
   - Weighted average of child node impurities
3. Compute Information Gain:
   Gain = B - A
4. Choose the attribute with the highest gain

Example using Entropy

Id	Terrain	Unicycle Type	Weather	Go for Ride?
1	Trail	Normal	Rainy	No
2	Road	Normal	Sunny	Yes
3	Trail	Mountain	Sunny	Yes
4	Road	Mountain	Rainy	Yes
5	Trail	Normal	Snowy	No
6	Road	Normal	Rainy	Yes
7	Road	Mountain	Snowy	Yes
8	Trail	Normal	Sunny	No
9	Road	Normal	Snowy	No
10	Trail	Mountain	Snowy	Yes

1. Compute impurity before splitting

- $B = \{Y^6, N^4\}$
- $E(B) = -p(Y)\log_2 p(Y) - p(N)\log_2(N)$
  $= -6/10 \log_2(6/10)$
  $= 0.44 + 0.53 = 0.97$

2. Compute Gain of each attribute

→ Terrain

1) Trail $= \{Y^2, N^3\}$

Entropy $= -\sum_{i=0} pi(t)\log_2 pi(t)$

$E(\text{Trail}) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5)$
$= 0.53 + 0.44$
$= 0.97$

2) Road = $\{Y^4, N^1\}$

$E(\text{Road}) = -4/5 \log_2(4/5) - 1/5 \log_2(1/5)$

$= 0.26 + 0.46 = 0.72$

3) Average Entropy of Terrain

$E(\text{Terrain}) = (5/10) E(\text{Trail}) + (5/10) E(\text{Road})$

$= 0.485 + 0.36 = 0.845$

$\Rightarrow \text{Gain}(S, \text{Terrain}) = E(S) - E(\text{Terrain})$

$= 0.97 - 0.845$

$= 0.125$

→ Unicycle_Type

1) Normal = $\{Y^2, N^4\}$

$E(\text{Normal}) = -2/6 \log_2(2/6) - 4/6 \log_2(4/6)$

$= 0.53 + 0.39 = 0.92$

2) Montain = $\{Y^4, N^0\}$

$E(\text{Montain}) = 0$ (maximal homogeneity → minimal impurity → minimal entropy)

3) Average Entropy

$E(U\_T) = (6/10) E(\text{Normal}) + (4/10) E(\text{Montain})$

$= 0.552$

$\Rightarrow \text{Gain}(S, \text{Unicycle\_Type}) = E(S) - E(U\_T)$

$= 0.97 - 0.552$

$= 0.418$

→ Weather

1) Rainy : $\{Y^2, N^1\}$

$E(\text{Rainy}) = -2/3 \log_2(2/3) - 1/3 \log_2(1/3)$

$= 0.39 + 0.53$

$= \boxed{0.92}$

2) Sunny : $\{Y^2, N^1\}$

$E(\text{Sunny}) = 0.92$

3) Snowy : $\{Y^2, N^2\}$

$E(\text{Snowy}) = \boxed{1}$ (minimal homogeneity → maximal impurity → maximal entropy)

4) Average Entropy

$E(\text{Weather}) = (3/10)E(\text{Rainy}) + (3/10)E(\text{Sunny}) + (4/10)E(\text{Snowy})$

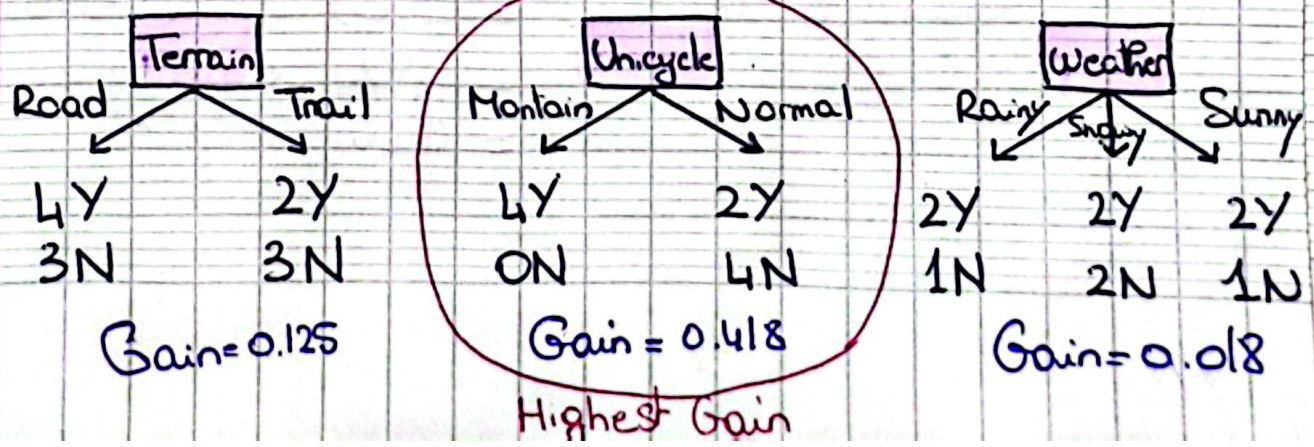
$= \boxed{0.952}$

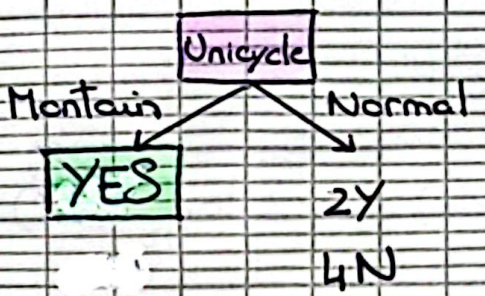
⇒ $\text{Gain}(S, \text{Weather}) = E(B) - E(\text{Weather})$

$= 0.97 - 0.952$

$= 0.018$

3. Choose the attribute with the highest gain





Base case: All data belong to the same class, stop and create a leaf node with that label

Id	Terrain	Unicycle	Weather	Go For Ride?
1	Trail	Normal	Rainy	No
2	Road	Normal	Sunny	Yes
5	Trail	Normal	Snowy	No
6	Road	Normal	Rainy	Yes
8	Trail	Normal	Sunny	No
9	Road	Normal	Snowy	No

Our new dataset S filtered on "Normal"

→ Before Splitting

$B = \{Y^2, N^4\}$

$E(B) = -2/6 \log_2 (2/6) - 4/6 \log_2 (4/6)$

$= 0.53 + 0.39$

$= 0.92$

→ Terrain

1) $Trail = \{Y^0, N^3\}$

$E(Trail) = 0$

2) $Road = \{Y^2, N^1\}$

$E(Road) = -(2/3) \log_2 (2/3) - (1/3) \log_2 (1/3)$

$= 0.39 + 0.53 = 0.92$

3) Average Entropy

$E(Terrain) = (3/6) E(Trail) + (3/6) E(Road)$

$\Rightarrow Gain(S, Terrain) = E(B) - E(Terrain)$

$= 0.92 - 0.46$

$= 0.46$

→ Weather

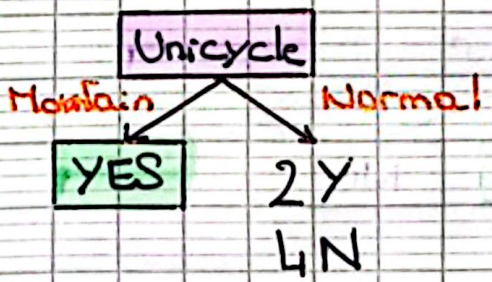
1) Rainy = {Y¹, N¹}
   E(R) = 1

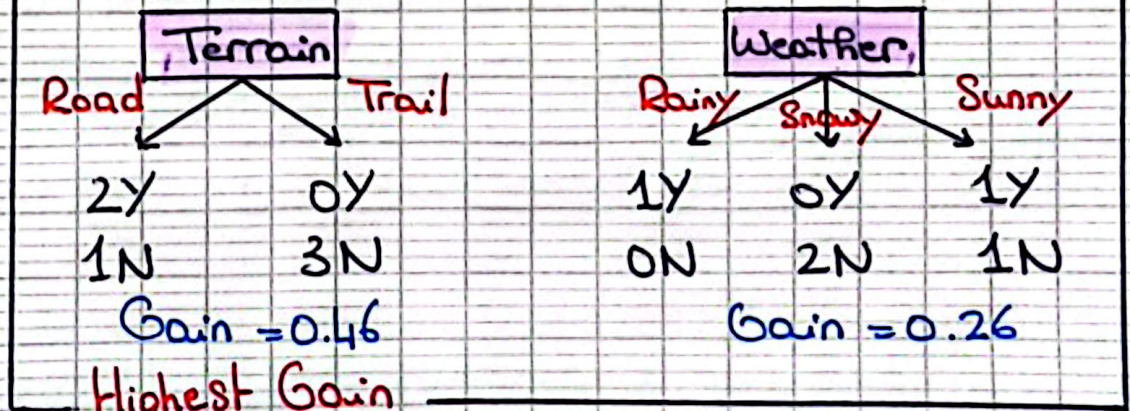
2) Sunny = {Y¹, N¹}
   E(Sunny) = 1

3) Snowy = {Y⁰, N²}
   E(Snowy) = 0

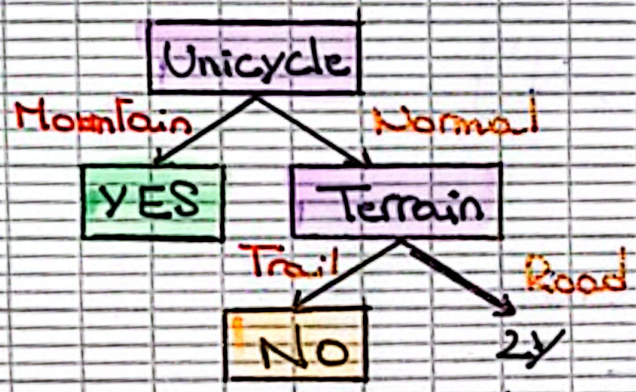
↳ Average Entropy
E(Weather) = 4/6 = 0.66

⇒ Gain (S, Weather) = E(B) − E(Weather)
= 0.92 − 0.66
= 0.26





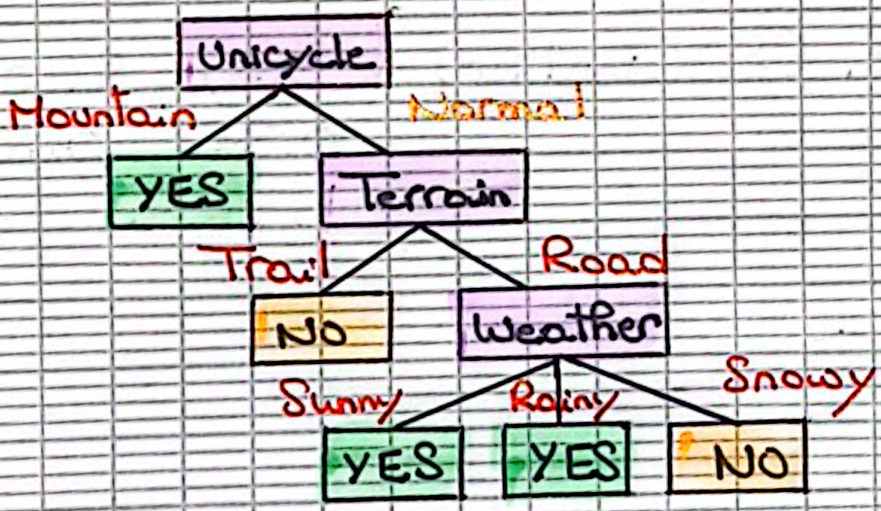
Highest Gain
↳ Split on Terrain



Id.	Terrain	Unicycle	Weather	Go for Ride?
2	Road	Normal	Sunny	YES
6	Road	Normal	Rainy	YES
9	Road	Normal	Snowy	No

Our new dataset 'S' Filtered on 'Normal' and 'Road'

=> The Final Decision Tree Model



↳ Example Using Gini_Index

Family = F
Sport = S
None = N

Id.	Income	No. of Child.	Car
1	Low	2 or more	Family
2	Low	2 or more	Family
3	Low	2 or more	Family
4	Low	2 or more	Family
5	Medium	2 or more	Family
6	Medium	2 or more	Family
7	High	2 or more	Family
8	High	2 or more	Family
9	High	0	Sport
10	High	0	Sport
11	High	1	Sport
12	High	1	Sport
13	Medium	0	None
14	Medium	0	None
15	Low	2 or more	None
16	Low	2 or more	None

. Compute Impurity before Splitting

$B = \{F^8, S^4, N^4\}$

$G.I(B) = 1 - [(8/16)^2 + (4/16)^2 + (4/16)^2]$

$= 0.625$

. Compute G.I for each attribute

↳ Income

1) Low: $\{F^4, S, N^2\}$

$G.I(L) = 1 - [(4/6)^2 + (2/6)^2]$

$= 0.44$

2) Medium: $\{F^2, S^0, N^2\}$

$G.I(M) = 1 - [(2/4)^2 + (2/4)^2]$

$= 0.5$

3) High: $\{F^2, S^4, N^0\}$

$G.I(H) = 1 - [(2/6)^2 + (4/6)^2] = 0.44$

Gini_Index $= 1 - \sum_j p_j^2$

↳ Average (Income) $= 6/16\,(0.44) + 4/16\,(0.5) + 6/16\,(0.44)$

⇒ $G.I(\text{Income}) = G.I(B) - G.I(\text{Income})$

$= 0.625 - 0.455$

$= 0.175$

→ Nb. of Children

1) 2 or more : $\{F^8, S^0, N^2\}$

G.I(2 or m) = $1 - [(8/10)^2 + (2/10)^2] = 0.32$

2) 0 : $\{F^0, S^2, N^2\}$

G.I(0) = $1 - [(2/4)^2 + (2/4)^2] = 0.5$

3) 1 : $\{F^0, S^2, N^0\}$

G.I(1) = $1 - [(2/2)^2] = 0$

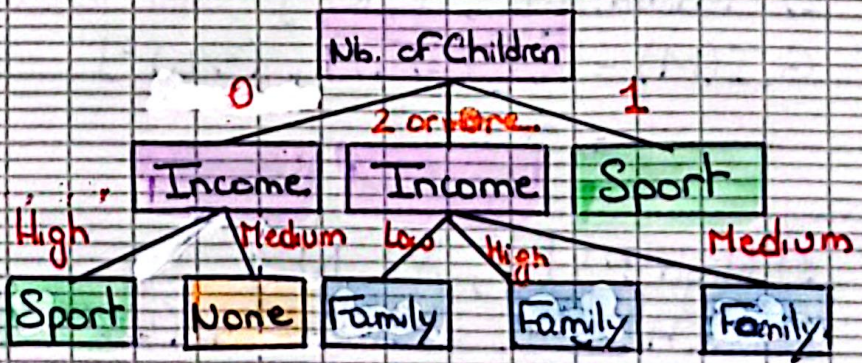
↳ Avg (Nb. of Children) = $10/16\,(0.32) + 4/16\,(0.5) + 2/16\,(0)$

$= 0.325$

⇒ G.I(Nb. of Children) = $0.625 - 0.325$

$= 0.3$

⇒ Nb. of Children has the higher G.I. so we split on it first



Example: Computing Gini Index for Continuous Attributes

1. Sort Income in increasing order
2. Find the midpoint between each pair of values $(a_i, a_{i+1})/2$
3. Choose the split position that has the least G.I or E

Id	Home owner	Marital Status	Income	Defaulted
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

	Annual Income																	
	60		70		75		85		90		95		100		120		125		220
	65		72		80		87		92		97		110		122		172	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	
YES	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	
NO	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	
Gini Index	0.396		0.374		0.34		0.414		0.4		0.3		0.34		0.37		0.396		

↳ minimum so we Split on (97)

$GI(<=65) = 1 - [(1/1)^2 + (0/1)^2] = 0$
$G.I(>65) = 1 - [(3/9)^2 + (6/9)^2] = 0.44$
} avg 1/10 x 0 + 9/10 x 0.44 = 0.396

$G.I(<=72) = 1 - [(0/2)^2 + (2/2)^2] = 0$
$G.I(>72) = 1 - [(3/8)^2 + (5/8)^2] = 0.468$
} Avg 0.3744

$G.I(<=80) = 1 - [(0/3)^2 + (3/3)^2] = 0$
$G.I(>80) = 1 - [(3/7)^2 + (4/7)^2] = 0.489$
} Avg 0.34

...