AI-Module-5-Important-Topics

- AI-Module-5-Important-Topics
  - 1. Forms of Learning
    - Supervised Learning
    - Classification
    - Regression
    - Unsupervised Learning
      - What it can do:
      - Clustering
    - Reinforcement Learning
    - Semi supervised learning
  - 2. Ockhams razor principle
    - Example
  - 3. Decision Tree

    - How It Works:
    - Key Points:
    - Uses:
    - Terminologies
    - Algorithm of decision tree
      - Example
    - Attribute Selection Measures (ASM)
      - Entropy
      - 1. Information Gain
      - 2. Gini Index
    - Pruning
    - Advantages of decison tree

1. Forms of Learning

Supervised Learning

- **What it is**: The machine learns using examples where the answers (labels) are already provided.
- **How it works**: These labels guide the machine to find patterns in the data.
- **Types of tasks**:
  1. **Classification**: Sorting things into categories (e.g., spam vs. not spam).
  2. **Regression**: Predicting numbers (e.g., house prices).

Classification

- **What it does**: The machine learns to sort data into specific categories.
- **How it works**: It identifies patterns in the data to decide how to label or define items.
- **Common methods**:
  - **Linear classifiers**: Simple decision boundaries.
  - **Decision trees**: Step-by-step decisions like a flowchart.

    - **Random forest**: Combines many decision trees for better accuracy.
- **Example**: Credit scoring:
    - Sorting customers as "low risk" or "high risk" based on their income and savings.

Regression

- **Purpose**: Understands the relationship between dependent (output) and independent (input) variables.
- **Uses**: Often for predictions, such as estimating future sales revenue.
- **Types**:
    - **Linear regression**: Straight-line relationship.
    - **Logistic regression**: Predicts probabilities (e.g., yes/no outcomes).
    - **Polynomial regression**: Fits data with curves, not just straight lines.
- **Example**
    - **Predicting House Prices**:
        - A regression model can predict the price of a house (dependent variable) based on factors like its size, number of bedrooms, and location (independent variables). For example, larger houses in prime locations tend to cost more.

Unsupervised Learning

Unsupervised learning is like giving the machine a puzzle without telling it what the final picture looks like. It works with raw, unlabeled data and tries to find hidden patterns.

What it can do:

1. **Clustering**: Grouping similar things together (e.g., sorting animals by size or habitat).
2. **Association**: Finding things that often appear together (e.g., people who buy bread also buy butter).
3. **Link prediction**: Spotting connections (e.g., suggesting friends on social media).
4. **Data reduction**: Simplifying data by keeping only the important bits (e.g., compressing an image).

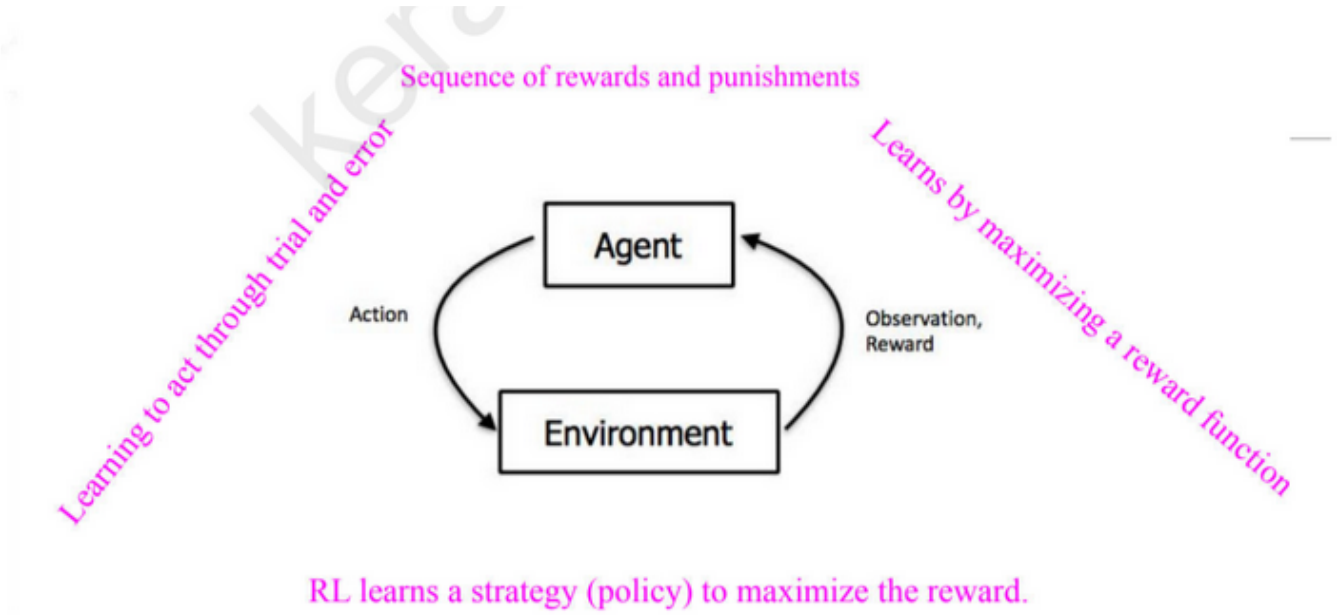
Clustering

Clustering is a way of organizing data into groups based on how similar or different they are.

Example:
Using **K-means clustering**, data points (like customer preferences) are grouped into clusters. The "K" value decides how many groups there will be.
Where it’s useful:

- **Market segmentation**: Grouping customers with similar buying habits.
- **Image compression**: Simplifying image details by grouping similar colors.

Reinforcement Learning



Reinforcement learning is like training a pet with rewards and punishments. The "agent" learns by trying actions and getting feedback.

Example 1: A taxi agent realizes it made a mistake if there’s no tip after a ride—it needs to figure out what went wrong.

Example 2: In a chess game, getting two points for a win teaches the agent it made good moves, but it has to figure out which moves led to the win.

The agent learns to improve by deciding which actions were most responsible for the results.

Semi supervised learning

In **semi-supervised learning**, we have a mix of labeled and unlabeled data. The system starts with a few labeled examples but has to figure out patterns from a much larger set of unlabeled ones.

Example:
Imagine you’re building a system to guess someone's age from a photo.

1. **Supervised learning**: You take pictures and ask people their age, creating labeled data (e.g., picture -> age).
2. **Reality**: Some people lie about their age, so the labels aren't perfect. The system has to handle the mix of correct and incorrect information, which introduces a bit of uncertainty.
3. **Challenge**: Unlabeled data and inaccurate labels make it tricky—this is where **semi-supervised learning** comes in, combining both labeled and unlabeled data to improve accuracy.

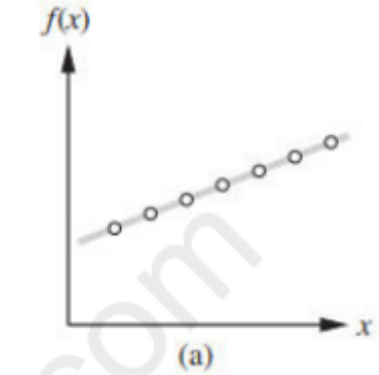
This creates a sort of "middle ground" between supervised and unsupervised learning, where the system works with both correct and noisy data.

2. Ockhams razor principle

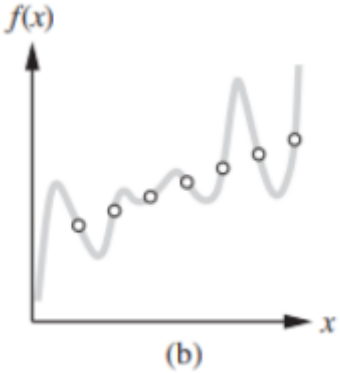
- Ockham's Razor is a principle that suggests **the simplest solution is often the best one**. This idea, named after the 14th-century philosopher William of Ockham, is used to solve problems where there are multiple possible explanations.
- When we try to find a function that fits some data, there might be many different ways to do it. Some solutions are simple, while others are more complex. **Ockham's Razor tells us to prefer the simpler solutions that fit the data**, as they are more likely to work well on new, unseen data

Example

- Imagine you have data points and you're trying to draw a line (or curve) to represent the relationship between the data points.
    - **A simple straight line (degree-1 polynomial)** fits the data well and is easy to understand.



- **A very complex curve (degree-7 polynomial)** fits the data perfectly but might be overcomplicating things. It might capture every small fluctuation in the data that isn't really important and could perform poorly on new data.



- In this case, Ockham's Razor suggests choosing the **simpler straight line** because it is more likely to generalize well to new data, rather than relying on the complex curve that might just be overfitting the current data.

3. Decision Tree

A **Decision Tree** is a tool used in **supervised learning** to solve two types of problems:

- **Classification:** Sorting things into categories (e.g., spam vs. not spam).
- **Regression:** Predicting numbers (e.g., house prices).

How It Works:

- Think of it like a flowchart:
  - **Nodes (circles):** Represent questions or conditions based on features of the dataset.
  - **Branches (lines):** Represent the possible answers or outcomes of the questions.
  - **Leaf Nodes (end points):** Represent the final decision or prediction.

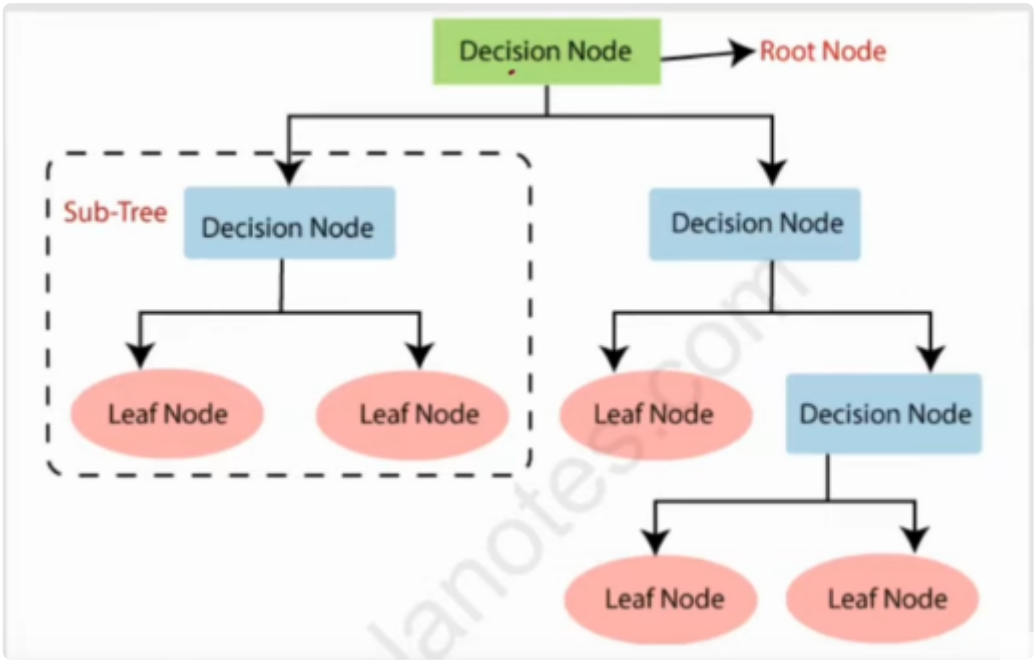
Key Points:

- Built using the **CART (Classification and Regression Tree)** algorithm.
- Decision Trees are easy to understand because they mimic how humans make decisions step-by-step.

Uses:

- Helps visualize and explain the logic behind decisions.
- Useful in applications where clear, interpretable decisions are important.

Terminologies

- 

    - Root Node
    - Leaf node
    - Splitting
    - Branch/Subtree
    - Pruning
    - Parent/child node

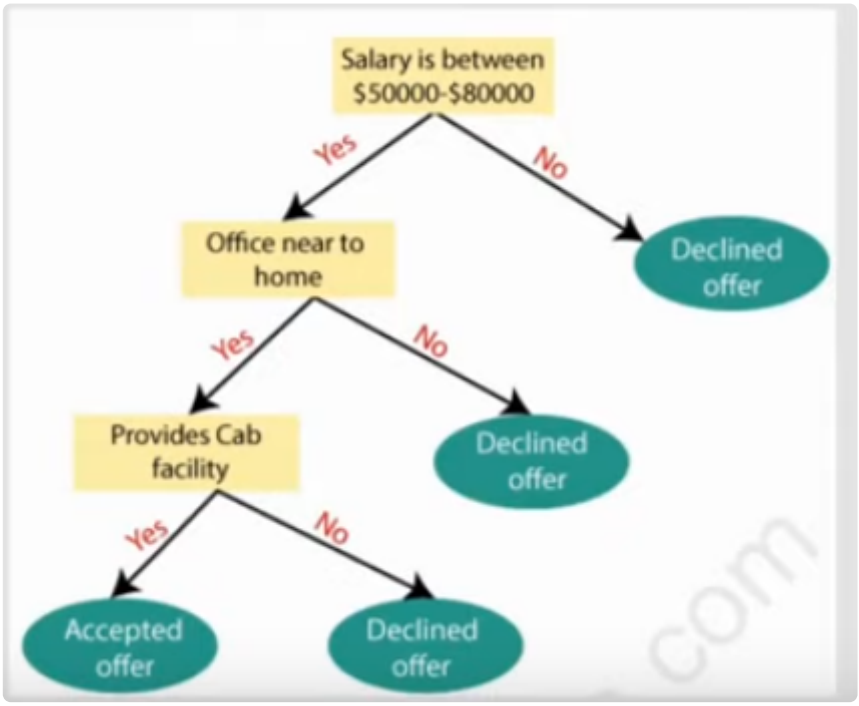
Algorithm of decision tree

- **1. Start with the Root Node**
    - Begin by selecting the entire dataset, denoted as **S**, and treat it as the root node.
- **2. Find the Best Attribute**

- Use an **Attribute Selection Measure (ASM)** (e.g., Gini Index, Information Gain, etc.) to identify the most important attribute for making a decision.

- **3. Split the Dataset**
  - Divide the dataset **S** into smaller subsets based on the possible values of the chosen attribute.
- **4. Create a Decision Node**
  - Add a decision node to the tree that represents the best attribute found in Step 2.
- **5. Repeat the Process Recursively**
  - For each subset created in Step 3, repeat Steps 2–4 to grow the tree further.
  - Stop when:
    - All data points in a subset belong to the same class, or
    - No more attributes are left to split on.
  - Mark the final nodes of the tree as **leaf nodes**, which represent the outcomes.

Example

- Suppose there is a candidate who has a job offer
- Decision should be made on whether he should accept the job or not
- 

- The Leaf nodes
  - It should indicate the end results.
  - Our end results here are "Accepted Offer" and "Declined offer"
- Our root node is the salary attribute

- Its split further on the condition whether "Office near to home"
- Its split further on condition "Provides Cab facilities"
- If all these conditions are satisfied then The job is accepted.

Attribute Selection Measures (ASM)

Lets understand what is entropy before going into ASM.

Entropy

- **Entropy** is a measure of **uncertainty** or **impurity** in the dataset. It helps us understand how mixed the data is before and after making a split in the decision tree.
- Entropy tells us how "messy" or "mixed" the data is.
  - If all data belongs to one class, entropy is **low** (0).
  - If data is equally mixed between classes, entropy is **high** (1 for two classes).
- **Example:**
  - Imagine we are classifying apples and oranges:
    - **Dataset 1:** All are apples → **Entropy = 0** (pure data).
    - **Dataset 2:** Half apples, half oranges → **Entropy = 1** (most mixed).

When building a decision tree, we need to decide which attribute (or feature) to split the data on at each step. **Information Gain** and **Gini Index** are methods used to find the best attribute for splitting.

1. Information Gain

- It tells us how much "confusion" (or uncertainty) is reduced when we split the data using an attribute.
- A good split makes the groups more **organized** (less mixed). The attribute that reduces the confusion the most is chosen.
- First, calculate the **Entropy**, which measures how mixed the data is.
- Then, calculate how much the Entropy decreases after splitting. This decrease is the **Information Gain**.
  Formula:
- $\text{Information Gain} = (\text{Entropy}) - (\text{Weighted Average Entropy})$
  Entropy Formula:
- $\text{Entropy} = -P(\text{yes}) \cdot \log_2 P(\text{yes}) - P(\text{no}) \cdot \log_2 P(\text{no})$

- **P(yes):** The chance of a "yes" outcome.
- **P(no):** The chance of a "no" outcome.

2. Gini Index

- It measures how **pure** or **impure** a split is.
- A good split creates groups that are as **pure** as possible (where most items in a group belong to the same class).
- A low **Gini Index** means the groups are clean and have little mixing of classes.
- The attribute with the lowest Gini Index is chosen.

Pruning

- Process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree
- We can use
  - Cost complexity Pruning
  - Reduced error pruning

Advantages of decison tree

- Simple to understand, useful for decision related problem, helpst to think about all possible outcomes

Disadvantages of decision tree

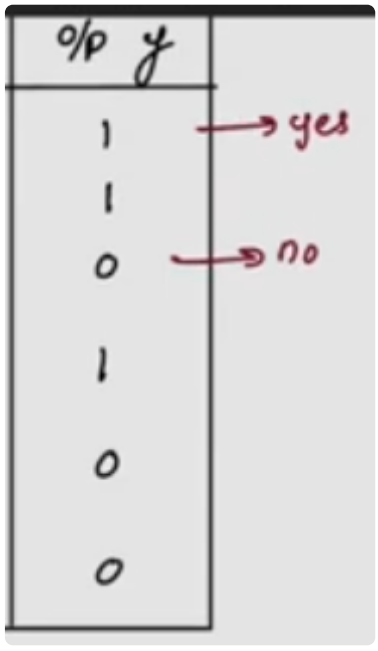
- Complex, overfitting issue
- Overfitting
  - A statistical model is said to be overfitted if it cant perform well with unseen data

4. Drawing Decision Trees

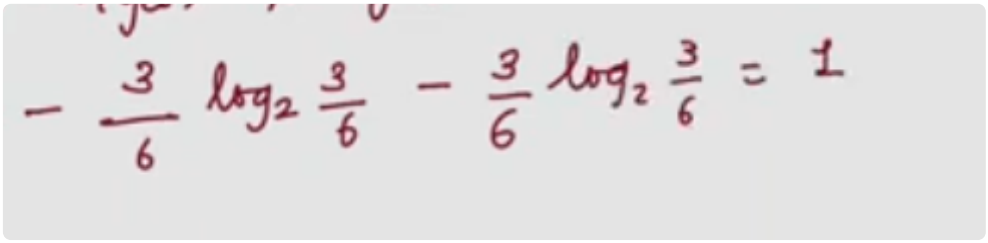
Consider the following dataset comprised of 2 binary input attributes (A1,A2) and 1 binary o/p. Use Decision Tree learning to learn decision tree from the data. Show the

computations made to determine the attribute to split at each node

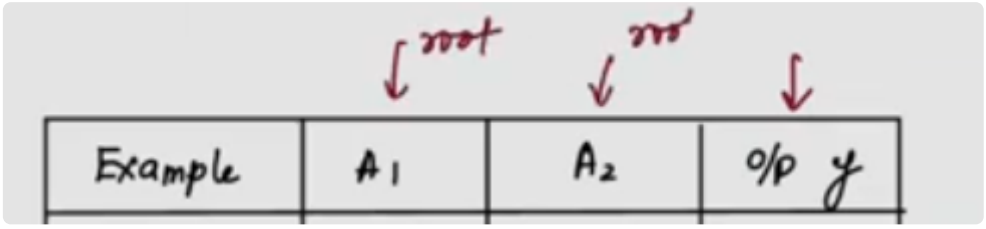
Example	A_1	A_2	o/p y
x_1	1	1	1
x_2	1	1	1
x_3	1	0	0
x_4	0	0	1
x_5	0	1	0
x_6	0	1	0

- First we need to select the Root node.
- We have to use ASM
- $\text{Information Gain} = (\text{Entropy}) - (\text{Weighted Average Entropy of each feature})$
- We need to find the entropy
  - $\text{Entropy} = -P(\text{yes}) \cdot \log_2 P(\text{yes}) - P(\text{no}) \cdot \log_2 P(\text{no})$
    - From the table, we need to get the P(yes) and P(no) values
    - 

    - 1 denotes Yes, and 0 denotes no
    - So the probabilities are

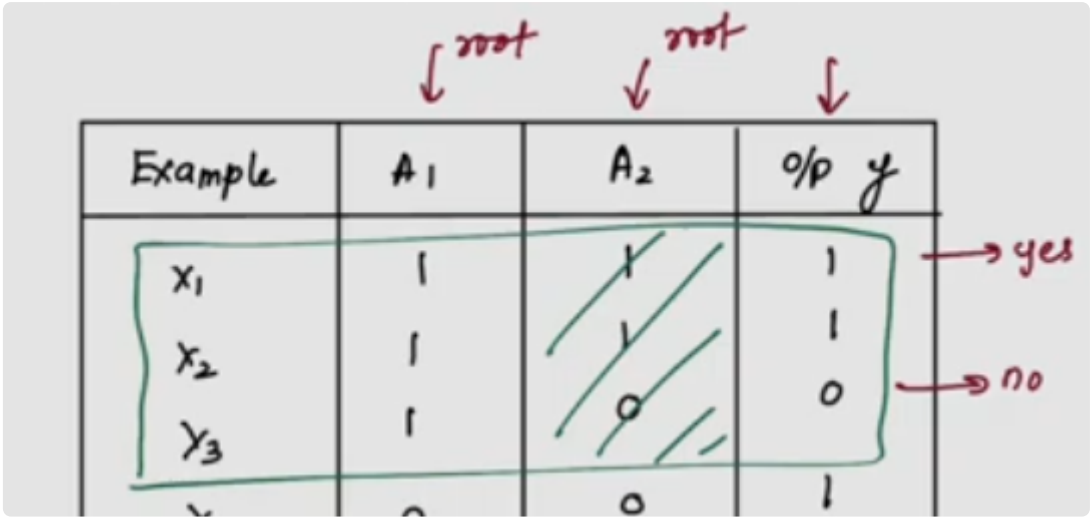
- P(yes) = 3/6
- P(no) = 3/6
- Subbing the values to the equation we get
- 

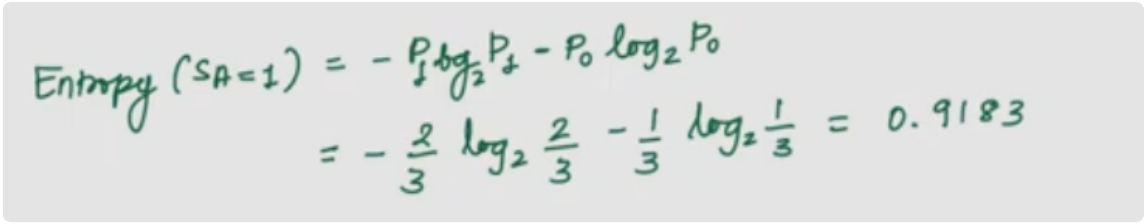
- Entropy = 1

- For choosing the root node, lets look at the table
- 

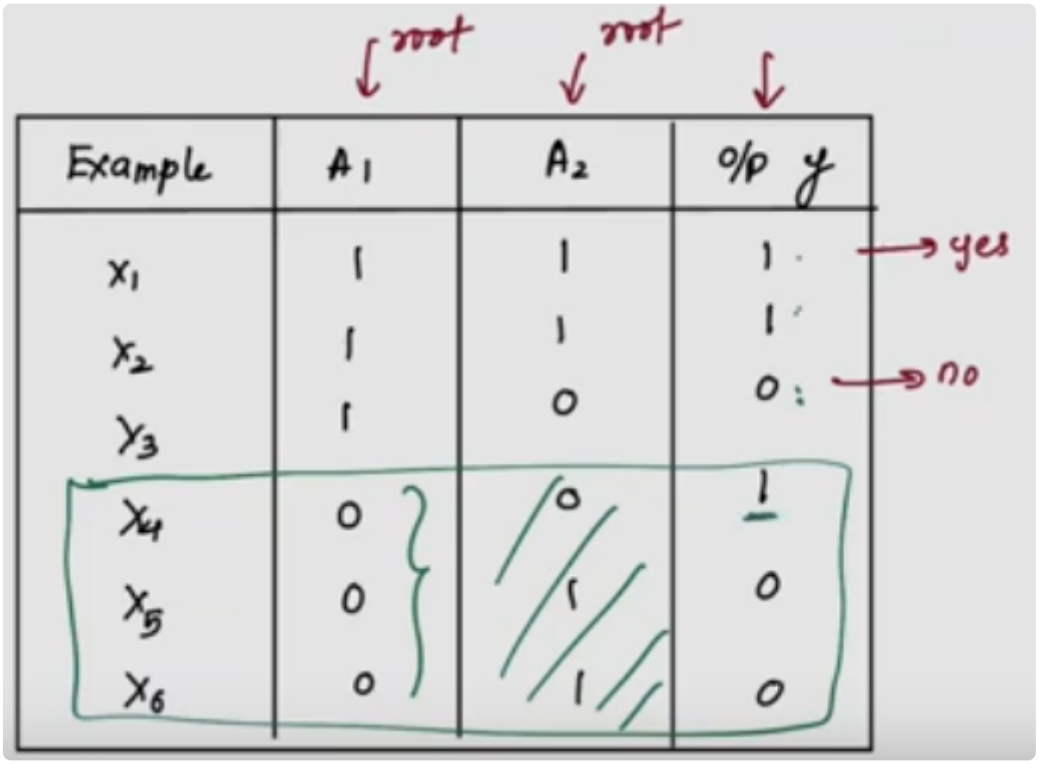
- Either A1 can be the root node or A2 can be the root node

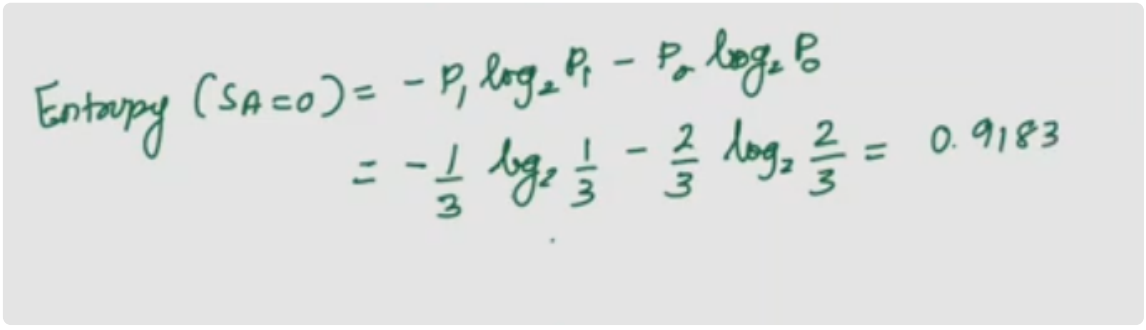
- Find information Gain for A1
- $\text{Gain}(A1) = \text{Entropy}(S) - \left(\frac{|S_{a1=1}|}{|S|} \cdot \text{Entropy}(S_{a1=1}) + \frac{|S_{a1=0}|}{|S|} \cdot \text{Entropy}(S_{a1=0})\right)$
  - $Gain(A1)$
    - **What it represents:** The Information Gain from splitting the dataset S using the attribute A1.
    - **Purpose:** Tells us how much uncertainty (or impurity) is reduced when we split S by A1
    - **Why it's important:** A higher Information Gain means A1 is a good candidate for splitting.
  - Splitting S by A1
    - When you split S based on A1, you create two subsets:
      - $S_{a1=1}$: Subset where $A1 = 1$.
      - $S_{a1=0}$ Subset where $A1 = 0$.
  - $\text{Entropy}(S_{a1=1})$
    - **What it represents:** Impurity in the subset $S_{a1=1}$.
    - **How to calculate:** Same entropy formula, but only for $S_{a1=1}$.
    - **Example:**
      For $S_{a1=1}$ (6 examples: 4 Class 1, 2 Class 0):
      - $p_1 = \frac{4}{6} = 0.667, p_0 = \frac{2}{6} = 0.333$

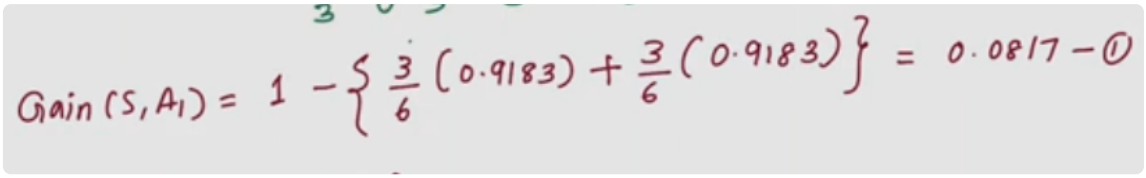
        - $\text{Entropy}(S_{a1=1}) = -\left(0.667 \cdot \log_2(0.667) + 0.333 \cdot \log_2(0.333)\right) = 0.918$
    - $\frac{|S_{a1=1}|}{|S|}$
        - **What it represents:** The fraction of examples in (S) that belong to $S_{a1=1}$
        - Example
            - $S_{a1=1}$ has 6 examples, (S) has 10 examples:
            - $\frac{|S_{a1=1}|}{|S|} = \frac{6}{10} = 0.6$
- Lets calculate Entropy of (Sa=1)
    - Observe the table Where A1=1, only A1 and Output columns
    - 

    - $\text{Entropy} = -P(1) \cdot \log_2 P(1) - P(0) \cdot \log_2 P(0)$
    - P(1)
        - In the output, there are two 1's and one 0
        - P(1) = 2/3
    - P(0)
        - In the output there is one 0 and two 1's
        - P(0) = 1/3
    - 

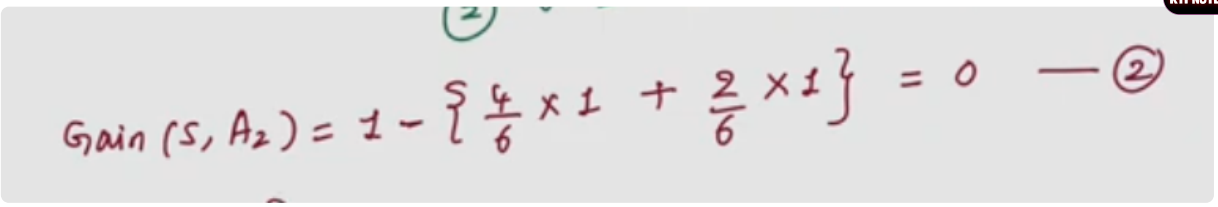
- Lets calculate Entropy of (Sa=0)
    - Observe the table where A1=0, Only A1 and output columns

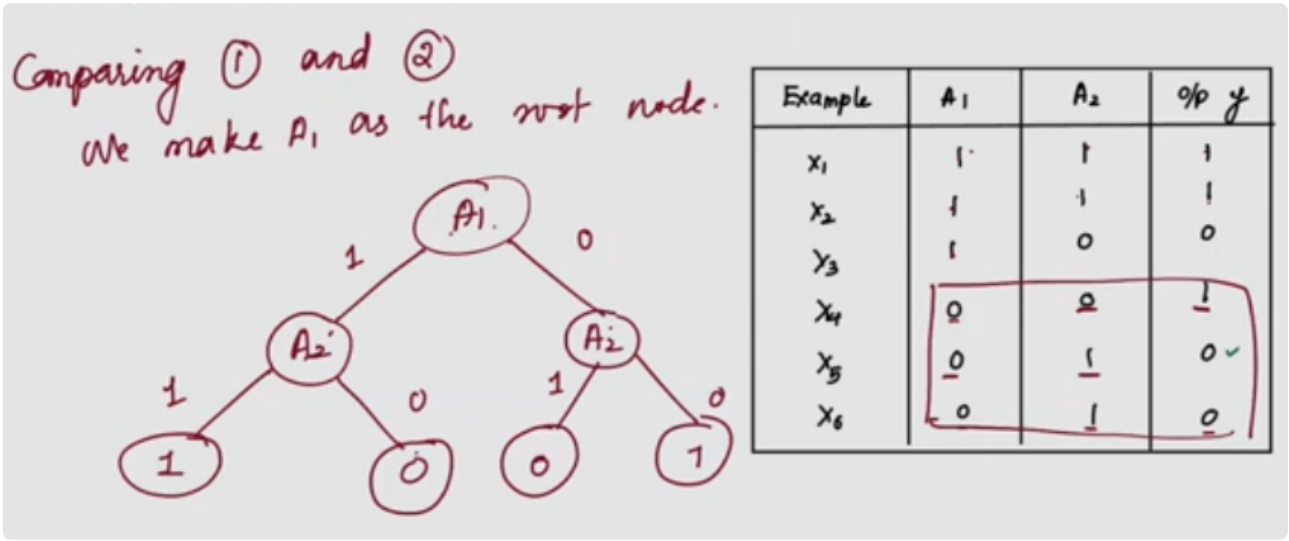
- 

- $\text{Entropy} = -P(1) \cdot \log_2 P(1) - P(0) \cdot \log_2 P(0)$
- P(1)
  - In the output, there are one 1 and two 0's
  - P(1) = 1/3
- P(0)
  - In the output there is two 0's and one 1
  - P(0) = 2/3
- 

- Subbing Values to get the Gain for A1
- 

- Find Information Gain for A2
  - Doing similar steps as before, we will get

- 

- Comparing Both Gains, Gain of A1 is larger, So we make **A1 as the root node**
  - When A1 = 1, A2 = 1. The output is 1
  - When A1 = 1, A2 = 0, Output is 0
  - When A1 = 0, A2=0, Output is 1
  - When A1 = 0, A2 = 1, Output is 0
- We can create a tree based on this
- 

*5. Hypothesis, Regression, Classification, Best fit, Model performance

Hypothesis in Machine Learning

- In machine learning, a **hypothesis** is a proposed explanation or assumption about how things work, based on limited evidence or assumptions. Think of it as an educated guess.
- Example:
  - Imagine you're trying to predict how much a house will cost based on its size. Your hypothesis is the equation that describes how the size of the house relates to the price.

Methods to find a possible hypothesis:

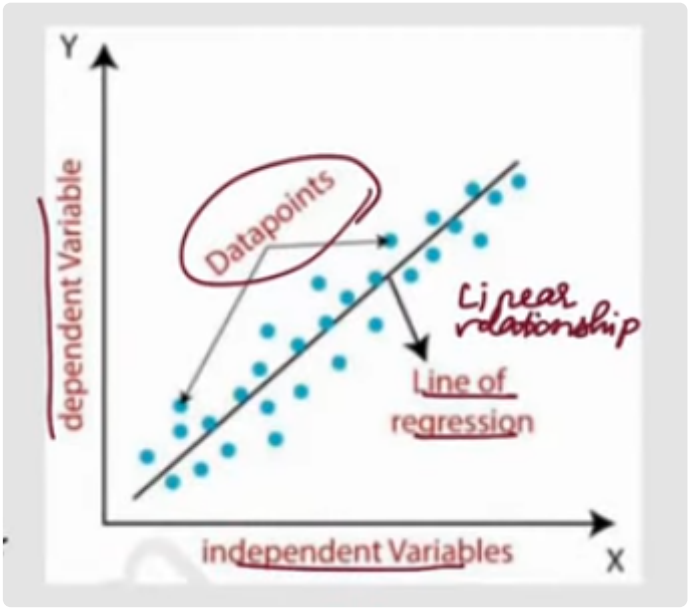
- **Hypothesis Space (H)**: This is the set of all possible hypotheses (or guesses) that could explain the relationship. In simple terms, it's all the different ways you could try to explain the problem.
- **Hypothesis (h)**: This is the specific hypothesis you choose that seems to best explain the data. It's like picking the best guess from all the possibilities in the hypothesis space.

Example of Hypothesis:

- A common way to represent a hypothesis is through a linear equation:
  - **y = mx + b**
    - **y** is the predicted value (for example, house price),
    - **m** is the slope of the line (how much the price changes as the size changes),
    - **x** is the input (the size of the house),
    - **b** is the y-intercept (the base price of the house when the size is zero).
- So, the hypothesis might say: "For each square meter of the house, the price increases by a certain amount."
- For instance, if the equation is:
  - **y = 100x + 50,000**,
- It means:
  - For each extra square meter (x), the price (y) increases by $100,
  - And the base price (b) of any house is $50,000 (even if the house has zero size).
  - This equation is the **hypothesis** that tries to describe the relationship between house size and price.

Regression and classification with linear models

- Linear regression: To make predictions for variables show a linear relationship between dependent variable and independent variables
- Based on regression line we can show

- 

    - The slope is represented by $y = a_0 + a_1x + \varepsilon$
        - y is dependent variable
        - x in independent variable
        - a0, a1 are the coefficients
        - ε -> Error
    - Positive Linear relationship: Y increases, X increases
    - Negative Linear relationship: Y decreases, X increases

Finding the best fit line

- Means the error between predicted values and actual values should be minimized.
- The best fit line will have least error.
- **Cost function**
    - Optimizes the regression coefficients or weights. It measures how a linear regression model is performing
    - $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (a_1x_i + a_0)\right)^2$
        - yi -> Actual value
        - $(a_1x_i + a_0))^2$ -> Predicted value
        - Difference between the actual and predicted value, will give us the error
    - To minimize MSE, we use gradient descent
        - Gradient Descent is used to update the coefficient of the line by reducing the cost function

Model Peformance

- Goodness of fit: Determines how the line of regression fits the set of observations. To find the best model, we can use the R-Squared method.
- R-squared method is a statistical method that measures the strength of relationship between dependent and independent variable on scale 0-100%
- The Highest R-squared value determines the less difference between the predicted value and actual value.
- R-squared = Explained Variation/ Total Variation

6. Univariate Linear Regression

- **Definition:**
  Univariate Linear Regression focuses on understanding the relationship between **one independent variable (input)** and **one dependent variable (output)**.
- **Formula:**
  - $y = a_0 + a_1 x + \epsilon$
  - (y): Dependent variable (what we're predicting)
  - (x): Independent variable (input feature)
  - (a_0): Intercept (value of (y) when (x = 0))
  - (a_1): Slope of the line (how much (y) changes for a unit change in (x))
  - (\epsilon): Error term (difference between actual and predicted values)
- **Example:**
  Suppose you want to predict the price of a product based on its weight:
  - (x): Weight of the product (in kg)
  - (y): Price of the product (in dollars)
  - Data:

Weight (kg)	Price ($)	
1	5	
2	10	
3	15	

The model might find a line like:

$y = 5x$

This means for every additional 1 kg of weight, the price increases by $5.

7. Multivariate Linear Regression

- **Definition:**
  Multivariate Linear Regression focuses on understanding the relationship between **multiple independent variables (inputs)** and **one dependent variable (output)**.
- **Formula:**
  - $y = a_0 + a_1x_1 + a_2x_2 + \ldots + a_nx_n + \epsilon$
  - y: Dependent variable (what we're predicting)
  - (x_1, x_2, ..., x_n): Independent variables (inputs)
  - (a_0): Intercept
  - (a_1, a_2, ... a_n): Coefficients (effect of each independent variable on (y))
  - ϵ: Error term
- **Example:**
  Suppose you want to predict the price of a house based on multiple factors:
  - (x_1): Size of the house (in square feet)
  - (x_2): Number of bedrooms
  - (x_3): Distance to the nearest city (in miles)
  - (y): Price of the house (in dollars)

Data

Size (sq ft)	Bedrooms	Distance (miles)	Price ($)
2000	3	5	300,000
2500	4	3	400,000
1800	2	8	250,000

The model might find an equation like:

- $y = 100x_1 + 50,000x_2 - 10,000x_3 + 20,000$

- For every additional square foot (x_1), the price increases by $100.
- For every additional bedroom (x_2), the price increases by $50,000.
- For every additional mile away from the city (x_3), the price decreases by $10,000.

8. Linear regression

Question

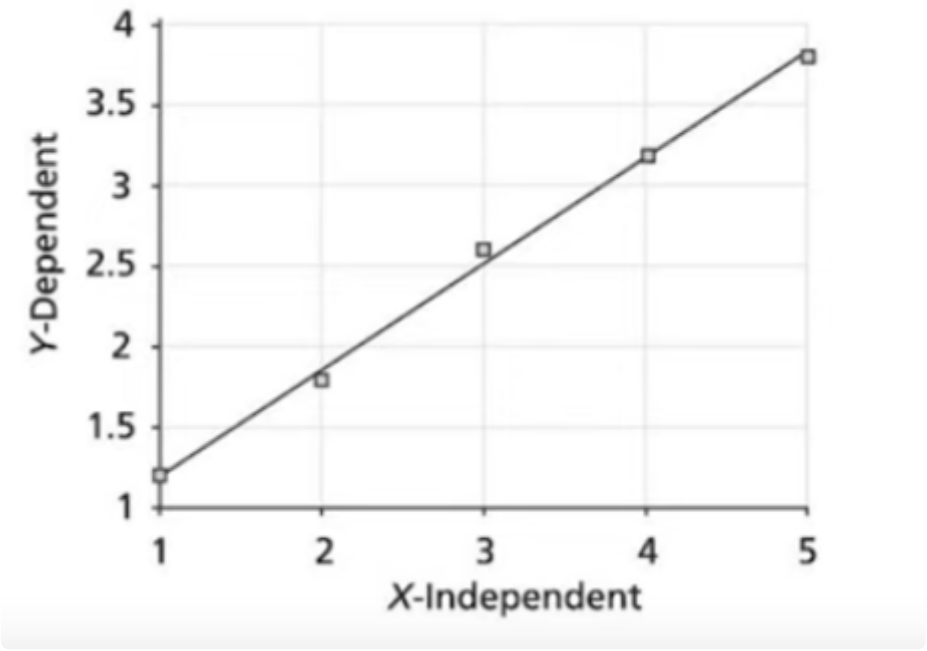
- Let us consider an example where 5 weeks sales data(in thousands) is given

x_i (Week)	y_j (Sales in Thousands)
1	1.2
2	1.8
3	2.6
4	3.2
5	3.8

-
- We need to apply linear regression technique to predict 7th and 12th week sales

Solution

1. First we will plot the dependent and independent variable



1.
    2. The goal of linear regression is to find a straight line which will fit the given dataset
2. Linear regression equation is given by
    1. $y = a_0 + a1 * x + e$

$$a_1 = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{\overline{x^2} - \bar{x}^2}$$

$$a_0 = \bar{y} - a_1 * \bar{x}$$

    2.
3. Finding the values

	x_i (Week)	y_j (Sales in Thousands)	x_i^2	$x_i * y_j$
	1	1.2	1	1.2
	2	1.8	4	3.6
	3	2.6	9	7.8
	4	3.2	16	12.8
	5	3.8	25	19
Sum	15	12.6	55	44.4
Average	$\bar{x} = 3$	$\bar{y} = 2.52$	$\overline{x^2} = 11$	$\overline{xy} = 8.88$

    1.
4. Putting the values into the equation

- $\bar{x} = 3 \qquad \bar{y} = 2.52 \qquad \overline{x^2} = 11 \qquad \overline{xy} = 8.88$

- $a_1 = \frac{(\overline{xy})-(\bar{x})(\bar{y})}{\overline{x^2}-\bar{x}^2} = \frac{8.88-3*2.52}{11-3^2} = 0.66$
- $a_0 = \bar{y} - a_1 * \bar{x} = 2.52 - 0.66 * 3 = 0.54$
- **Regression equation is**
- $y = a_0 + a_1 * x$
- $y = 0.54 + 0.66 * x$

1.

5. Our questions were to predict 7th and 12th week sales
6. 7th week means, Value of X = 7

The predicted 7th week sale (when x = 7) is,

y = 0.54 + 0.66 x 7 = 5.16

1.
2. So the sales will be 5.16

7. 12th week means, value of X = 12

the predicted 12th week sale (when x = 12) is,

1. **y = 0.54 + 0.66 x 12 = 8.46**

9. Classification using logistic regression

- **Logistic regression** is suitable for binary classification problem
- **Classification problems**
  - Is mail spam or not spam?
    - The answer is yes or no. Thus categorical dependent variable is a binary response of yes or no

- If the student should be admitted or not based on entrance examination marks
  - Here categorical variable response is admitted or not.
- The student being pass or fail is based on marks secured

How does logistic regression algorithm work?

- Consider the following example
  - An organization wants to determine an employees salary increased based on their performance
  - For this purpose, a linear regression algorithm will help them decide
  - Plotting a regression line by considering the employees performance as the independent variable and the salary increase as the dependent variable will make their task easier.

  - Here the red line is the linear regression line, from the line we can figure out the salary hike
  - For eg: If the employee rating is 3, then the salary hike will be around 15
- Now suppose, the organisation wants to know whether an employee would get a promotion or not based on their performance
  - The previous linear graph wont be suitable in this case
  - We need to clip the line at 0 and 1, and convert to a sigmoid curve. (S curve)
  - Based on the probability values the organization can decide whether an employee will get a salary increase or not.

- Lets set the Threshold for promotion to 0.5

  - Here, if employee rating is 4, he has probability of 0.75, which is greater than the threshold, so he will be promoted