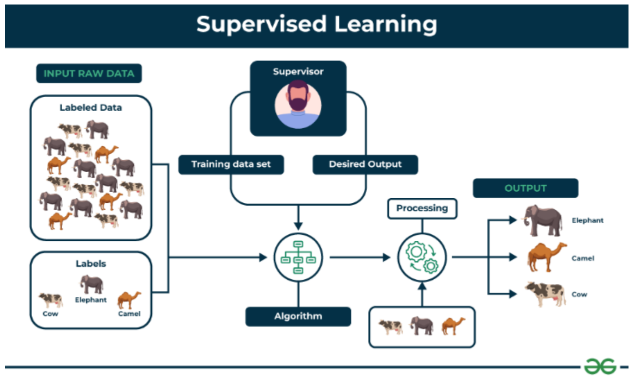
Q1. Differentiate between the Supervised and Unsupervised Learning.

<u>Solution of Q1:</u>

Supervised and unsupervised learning are two primary types of machine learning paradigms, each with distinct goals, approaches, and applications. Here's a comparison:

Aspect	Supervised Learning	Unsupervised Learning
Definition	Learning from labeled data where the target/output is provided.	Learning patterns or structure from unlabeled data without a target.
Data	Requires labeled datasets with input-output pairs.	Works with unlabeled datasets (only inputs, no predefined outputs).
Objective	Predict or classify based on the relationship between inputs and outputs.	Find hidden patterns, structures, or groupings in the data.
Output	Predictions (regression or classification) or mapped relationships.	Clusters, associations, or reduced feature representations.
Common Algorithms	- Linear Regression- Logistic Regression- Decision Trees- Support Vector Machines (SVM)- Neural Networks	- K-Means Clustering- Hierarchical Clustering- Principal Component Analysis (PCA)- Autoencoders- DBSCAN
Applications	- Spam detection- Credit scoring- Image recognition- Sentiment analysis- Forecasting	- Customer segmentation- Market basket analysis- Dimensionality reduction- Anomaly detection

- **Supervised Learning**: Focuses on prediction using labeled datasets. It is like a "teacher" guiding the learning process e.g. A teacher provides correct answers (labels) during training and evaluates the model's predictions on new data.
- **Unsupervised Learning**: Explores the dataset to identify patterns or groupings without explicit guidance, working autonomously e.g. There is no teacher; the model must find its own patterns or structures in the data.



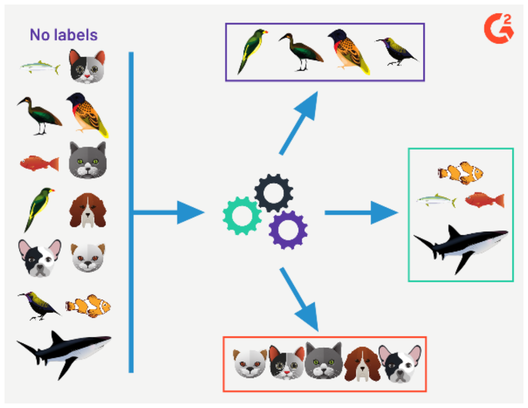


Fig: Supervised Learning   Fig: Unsupervised Learning

Q2. Explain how gradient descent is used in logistic regression for optimizing model parameters. Include the steps involved, the cost function used, and how the parameters are updated during the process.

<u>Solution of Q2:</u>

<u>Cost Function in Logistic Regression</u>:

We utilize the sigmoid function (or logistic function) to map input values from a wide range into a limited interval. Mathematically, the sigmoid function is:

$$\boldsymbol{\sigma(t) = \frac{1}{1+e^{-t}}}$$

We define the logistic regression model for binary classification problems. We choose the hypothesis function to be the sigmoid function:

$$\boldsymbol{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$$

The cost function for logistic regression is defined as:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)]$$

m : Number of training samples.

$y^{(i)}$: True label for the iii-th training example (0 or 1).

$h_\theta\left(x^{(i)}\right)$: Model's predicted probability for the i-th sample.

<u>Steps in Gradient Descent for Logistic Regression:</u>

Start with initial values for parameters θ = $[\theta_0, \theta_1, \ldots \ldots .. \theta_n]$ usually set to zero or small random values.

Compute the Gradient:

The gradient of the cost function with respect to each parameter θ_j is calculated using:

$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(\mathrm{h}_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$

$x_j^{(i)}$ =The j-th feature of the i-th training example.

$\mathrm{h}_\theta(x^{(i)}) - y^{(i)}$ =The error between the prediction and the actual label.

Plugging this into the gradient descent function leads to the update rule:

$$\theta_j = \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(\mathrm{h}_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)}$$

Update the Parameters:

Using the computed gradient, update each parameter using the gradient descent rule:

$$\theta_j = \theta_j - \alpha\frac{\partial J(\theta)}{\partial \theta_j}$$

α: Learning rate, which controls the step size of updates.

This steps moves θ in the direction that minimizes the cost function .

Repeat Untill Convergence:

Repeat Compute the Gradient and Update the Parameters steps untill the cost function $J(\theta)\ converges.$

How Parameters are Updated:

- The logistic regression model uses batch gradient descent (processing the entire dataset per iteration), stochastic gradient descent (SGD) (updating parameters using one example at a time), or mini-batch gradient descent (updating parameters using small subsets of the dataset).

Q3. Consider a binary classification problem where you predict whether a student passes (1) or fails (0) based on study hours. Explain how this model produces a probability value and converts it into a binary outcome using Logistic Regression. You can use a sample dataset use the steps as in Q2.

Solution of Q3:

Logistic regression models the probability of a binary outcome using the sigmoid function. The sigmoid function maps any input (to a value between 0 and 1, which can be interpreted as a probability. The sigmoid function is given by:

$$\sigma(t) = \frac{1}{1+e^{-t}}$$

$$\sigma(t) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$$

$$P\ (\text{pass} \mid \text{study hours}) = \frac{1}{1+e^{-(\theta_0+\theta_1 . study\ hour)}}$$

where:

- P (pass | study hours) is the probability that the student passes given the study hours.
- θ_0 is the intercept (bias).
- θ_1 is the weight associated with study hours.

Sample Dataset

Study Hours	Result (1=Pass| 0=Fail)
2	0
4	0

5	1
6	1
8	1

(a)How to calculate the probability of pass for the student who studied 3 hours?

(b)How to calculate the probability of pass for the student who studied 10 hours?

Ans(a):

we can compute the probability of passing as follows:

1. Calculate the linear combination (logit):

$$\text{Logit(t)} = \theta_0 + \theta_1 . study\ hours$$

2. Plug the logit value into the sigmoid function to get the probability.

We know that:

1. If P(pass | study hours)≥0.5 the model predicts that the student will pass (output 1).
2. If P(pass | study hours)<0.5 the model predicts that the student will fail (output 0).

This threshold (0.5) is often chosen as a standard, but it can be adjusted depending on the application.

Assume the model parameters are:

- $\theta_0 = -3$
- $\theta_1 = 0.8$

If a student studies for 3 hours, the model computes:

$$\text{Logit}(t) = -3+0.8\times 3 = -0.6$$

$$P(\text{pass} \mid \text{study hours}=3) = \frac{1}{1+e^{-(-0.6)}}$$

$$=0.35$$

Since 0.35 is less than 0.5, the model predicts 0 (fail)

(b) If a student studies for 10 hours, the model computes:

$$\text{Logit}(t) = -3+0.8\times 10 = 5$$

$$P(\text{pass} \mid \text{study hours}=10)= \frac{1}{1+e^{-5}}$$

$$= 0.99$$

Since 0.99 is greater than 0.5, the model predicts **1** (pass)

Q4. A company wants to determine whether a customer should receive a promotional discount based on Age, Income, and Shopping Frequency. Create a simple dataset with 10 rows and build a decision tree based on Information Gain for the first two levels of the tree. Interpret the results.

<u>Solution of Q4:</u>

Age	Income	Shopping Frequency	Promotional Discount
<=30	Low	High	Yes
31…40	Medium	Medium	No
>40	High	Low	No
<=30	Medium	Medium	Yes
>40	Low	Low	No

31…40	High	High	Yes
<=30	Medium	High	Yes
>40	Medium	Medium	No
31…40	Low	Low	No
<=30	High	Medium	Yes

Attribute Selection Measure: Information Gain

$\boldsymbol{D}$ = set of data instances

$|\boldsymbol{D}|$ = number of data instances (10)

$\boldsymbol{p_i}$=the probability that an arbitrary tuple in $\boldsymbol{D}$ belongs to class $\boldsymbol{C_i}$

$$\boldsymbol{p_i = \frac{|C_{i,D}|}{|D|}}$$

Here, class yes is 1

$$\boldsymbol{p_1 = \frac{|C_{1,D}|}{|D|} = \frac{5}{10}}$$

$$\boldsymbol{p_2 = \frac{|C_{2,D}|}{|D|} = \frac{5}{10}}$$

Expected information (entropy) needed to classify a tuple in D:

$$\boldsymbol{Info(D) = -\sum_{i=1}^{m} p_i log_2 p_i}$$

For the given data

$$\boldsymbol{Info(D) = -\sum_{i=1}^{2} p_i log_2 p_i}$$

$$\boldsymbol{= -p_1 log_2 p_1 - p_2 log_2 p_2}$$

$$= -\frac{5}{10}\log\frac{5}{10} - \frac{5}{10}\log\frac{5}{10} = 1$$

Information needed (after using A to split D into v partitions)

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Example

$$Info_{Age}(D) = \sum_{j=1}^{3} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$= \frac{4}{10} \times I(4,0) + \frac{3}{10} \times I(1,2) + \frac{3}{10} \times I(0,3) = 0.275$$

$\frac{\mathbf{4}}{\mathbf{10}} \times \boldsymbol{I}(\mathbf{4},\mathbf{0})$ means "age <=30" has 4 out of 10 samples, with 4 yes'es and 0 no's

Similarly

$$Info_{Income}(D) = \sum_{j=1}^{3} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$= \frac{3}{10} \times I(1,2) + \frac{4}{10} \times I(2,2) + \frac{3}{10} \times I(2,1) = 0.95$$

Again

$$Info_{Shopping\ Frequency}(D) = \sum_{j=1}^{3} \frac{|D_j|}{|D|} \times Info(D_j)$$

$$= \frac{3}{10} \times I(0,3) + \frac{4}{10} \times I(2,2) + \frac{3}{10} \times I(3,0) = 0.4$$

We know,

$$\boldsymbol{Gain(A) = Info(D) - Info_A(D)}$$

$$Gain(Age) = Info(D) - Info_{Age}(D)$$

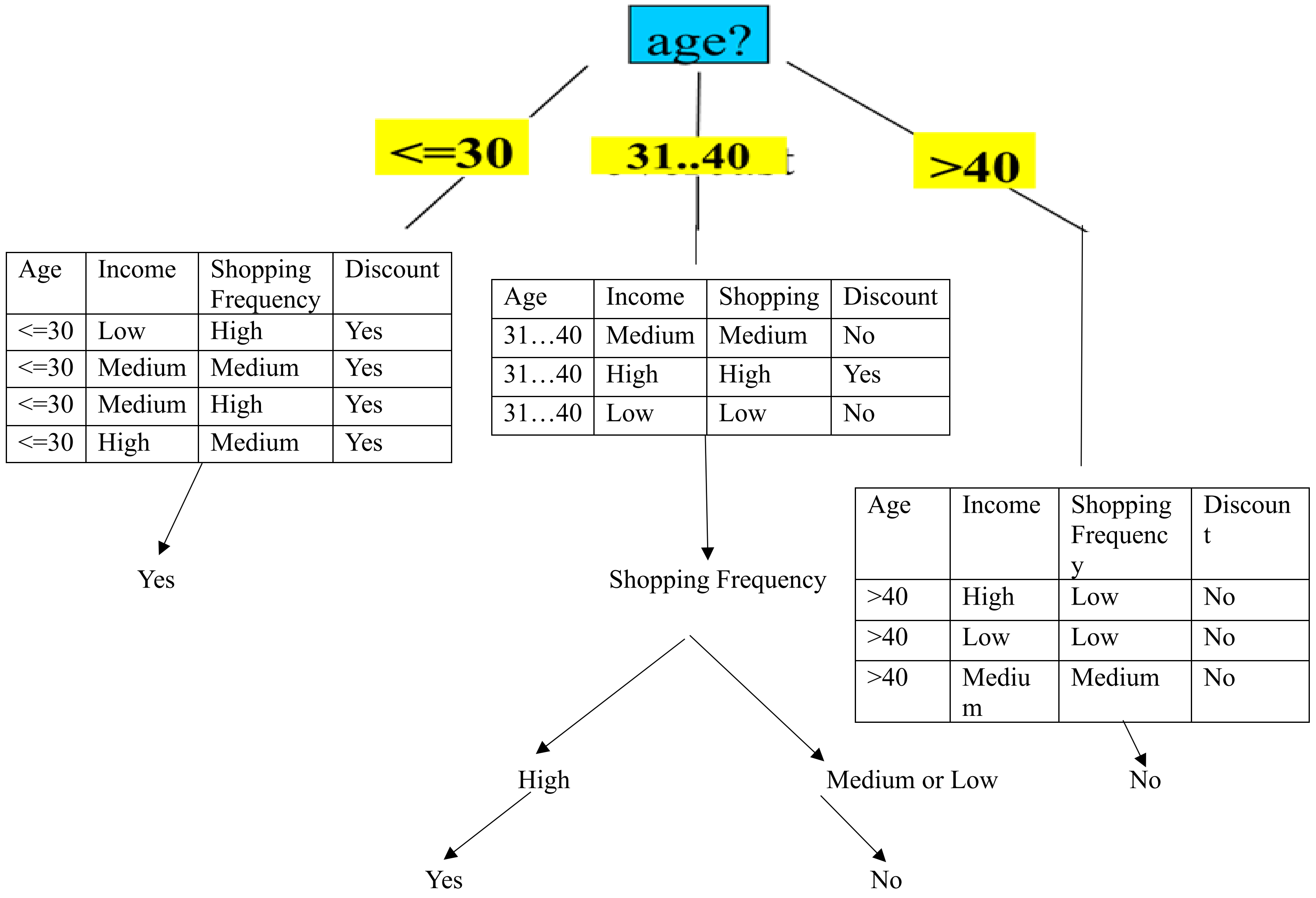
$$= 1 - 0.275 = 0.725$$

$$Gain(Income) = Info(D) - Info_{Income}(D)$$

$$= 1 - 0.95 = 0.05$$

$$Gain(Shopping\ Frequency) = Info(D) - Info_{Shopping\ Frequency}(D)$$

$$= 1 - 0.4 = 0.6$$



Age	Income	Shopping Frequency	Discount
<=30	Low	High	Yes
<=30	Medium	Medium	Yes
<=30	Medium	High	Yes
<=30	High	Medium	Yes

Age	Income	Shopping	Discount
31…40	Medium	Medium	No
31…40	High	High	Yes
31…40	Low	Low	No

Age	Income	Shopping Frequenc y	Discoun t
>40	High	Low	No
>40	Low	Low	No
>40	Mediu m	Medium	No

Solution of Q5:

The middle 5 digits of my id is 46052

So, AB=46

CDE=052=52

Data Table

	Predicted Spam	**Predicted Not Spam**
Actual Spam	46	10
Actual Not Spam	20	52

(a)

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$= \frac{46+52}{128}$$

$$= 0.766$$

$$precision = \frac{TP}{TP+FP}$$

$$= \frac{46}{46+20}$$

$$= 0.697$$

$$recall = \frac{TP}{TP+FN}$$

$$= \frac{46}{46+10}$$

$$= 0.821$$

$$F1 = \frac{2 \times precion \times recall}{precision + recall}$$

$$= \frac{2 \times 0.697 \times 0.821}{0.697 + 0.821}$$

$$= 0.754$$

So,

- Accuracy: 0.766 (or 76.6%)
- Precision: 0.697 (or 69.7%)
- Recall: 0.821 (or 82.1%)
- F1 Score: 0.754 (or 75.4%)

(b)

Accuracy: The model correctly predicted the outcome for about 76.6% of the emails, which is reasonable but not exceptional.

Precision: About 69.7% of the emails predicted as spam are actually spam. This indicates a moderate rate of false positives.

Recall: The model correctly identifies 82.1% of the actual spam emails. This high recall is useful for ensuring most spam is caught.

F1 Score: The balance between precision and recall suggests the model performs decently but has room for improvement, especially in reducing false positives.

Thus, the model prioritizes identifying spam (high recall), which is good for spam detection systems. However, the relatively lower precision indicates it could unnecessarily classify non-spam emails as spam. Improving precision would enhance the model's usability. Rather than that, the model has a good performance.