ACTIVITY – CASE STUDY

NAME: Harinie.S

REG.NO: 512223205013

COLLEGE: SKP ENGINEERING COLLEGE

COLLEGE CODE: 5122

1. SUPERVISED LEARNING:

Supervised learning is a type of machine learning where the model is trained using labeled data — meaning, the input data (X) already has a known output (Y).

The algorithm learns the relationship between inputs and outputs, so it can predict the output for new data.

Linear Regression:

Linear Regression is a supervised learning algorithm used for predicting continuous values.

It assumes a linear relationship between input variables (X) and the output variable (Y).

The general equation is:

$$Y = m X + c$$

Y = Dependent variable (what we want to predict)

X = Independent variable (input feature)

m = Slope (how much Y changes for each unit of X)

c = Intercept (value of Y when X = 0)

Real –Time Example: Predicting House Price

Size(sqft)	Price($)
1000	150,000
1500	200,000
1800	230,000
2400	275,000
3000	320,000

Problem Solution:

The company has historical data of house sizes and their corresponding prices.

Build a supervised learning model using linear regression to predict the price of a house when its size is given.

Solution Statement:

Step 1: Import the data collect and prepare the data (size vs. price).

Step 2: Train the model Use linear regression to fit the data:

$$Price = m \times Size + c$$

Step 3: Find the best-fit line the model calculates m and c using the least squares method:

$$m = \frac{N(\sum XY) - (\sum X)(\sum Y)}{N(\sum X^2) - (\sum X)^2}$$

$$c = \frac{(\sum Y) - m(\sum X)}{N}]$$

Step 4: Make predictions For a house of size 2000 sq.ft, use the equation:

$$Predicted\ Price = m(2000) + c$$

Step 5: Evaluate performance Check how well the model performs using metrics like:

Mean Squared Error (MSE)

R^2 Score (Coefficient of Determination)

Example Output:

If the model finds:

$$m = 100,\ c = 50,000$$

$$Price = 100 \times Size + 50,000$$

For a 2000 sq.ft house:

$$Price = 100(2000) + 50,000 = 250,000$$

So, the model predicts a $250,000 house price.

2. UNSUPERVISED LEARNING:

Unsupervised learning is a type of machine learning where the algorithm is given unlabeled data — meaning, the inputs are known but there are no predefined outputs.

The model tries to find hidden patterns or groupings within the data on its own.

K-Means Clustering:

K-Means Clustering is one of the most popular unsupervised learning algorithms used for grouping data into K distinct clusters based on their similarity.

It works by:

1. Choosing the number of clusters (K).

2. Randomly assigning K centroids (points representing each cluster).

3. Assigning each data point to the nearest centroid (based on distance, usually Euclidean).

4. Updating centroids by taking the average position of points in each cluster.

5. Repeating steps 3–4 until the centroids no longer move significantly (convergence).

Mathematical Formula:

The goal is to minimize the within-cluster sum of squares (WCSS):

$$\text{WCSS} = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

= cluster

= centroid of cluster

= data point

Real –Time Example:

Example: Customer Segmentation in a Shopping Mall

A mall owner wants to understand customer behavior.

He has data for each customer such as:

Annual Income

Spending Score (1–100)

He doesn't know how many types of customers exist.

Using K-Means Clustering, he can group customers into clusters like:

High Income, High Spending

Low Income, Low Spending

High Income, Low Spending

Moderate Income, Moderate Spending

This helps in targeted marketing and decision-making.

Problem Solution:

To group a set of data points into K distinct clusters based on similarity, so that points in the same cluster are more similar to each other than to those in other clusters.

Solution Statement:

Step 1: Data Preparation Collect data such as:

| Customer ID | Annual Income ($k) | Spending Score (1–100) || 1 | 15 | 39 | | 2 | 16 | 81 | | 3 | 17 | 6 | | 4 | 18 | 77 | | 5 | 19 | 40 | | ... | ... | .. |

Step 2: Choose Number of Clusters (K)

Use the Elbow Method to find the optimal K:

Plot WCSS vs. K.

The point where the WCSS starts to flatten ("elbow point") indicates the optimal number of clusters.

Step 3: Apply K-Means Algorithm

Initialize K centroids randomly.

Assign each data point to its nearest centroid.

Recalculate the centroid of each cluster.

Repeat until centroids stabilize.

Step 4: Visualize the Clusters Plot the clusters using 2D scatter plots — income vs. spending score — each cluster shown in different colors.

Step 5: Interpret Results Example:

| Cluster | Characteristics | Marketing Strategy | | 1 | High Income, High Spending | Premium offers | | 2 | Low Income, Low Spending | Discount products | | 3 | Moderate Income, Moderate Spending | Loyalty programs

Example Output Visualization:

A 2D scatter plot might show 5 clusters like:

Cluster 1: High income, high spending

Cluster 2: High income, low spending

Cluster 3: Low income, high spending

Cluster 4: Low income, low spending

Cluster 5: Moderate income, moderate spending

3.REINFORCEMENT LEARNING(RL):

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment.

The agent receives rewards (positive or negative) based on its actions and learns the best strategy (called a policy) to maximize the total reward over time.

Q-Learning:

Q-Learning is a model-free reinforcement learning algorithm — meaning the agent does not need to know the dynamics of the environment (like transition probabilities).

It learns a function called the Q-value (or action-value), denoted as: Q(s, a)

which estimates the expected future reward of taking action a in state s and following the optimal policy afterward.

Q-Learning Update Rule

$$Q(s, a) = Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

Where:

Symbol	**Meaning**
S	Current state
A	action taken
R	Immediate reward received
s’	next state
a’	next action
A	learning rate ($0<a\leq1$)
Y	discount factor($0\leq v<1$)
max_{a’}Q(s’,a’)	estimated best future reward from next state

Real-Time Example: Self-Driving Car at a Traffic Signal

Environment:

A self-driving car approaches a traffic light.

Possible States (s):

Red Light

Yellow Light

Green Light

Possible Actions (a):

- Stop
- Go
- Slow Down

Rewards (r):

The car starts learning over many episodes (trials).

It tries different actions in each state and updates the Q-values according to the received rewards.

Eventually, it learns the optimal policy:

Stop at red light

Go at green light

Slow down at yellow

Situation	Reward
Go on green	+10
Stop on red	+5
Go on red	-20
Stop on green	-5
Slow down on yellow	+2

Problem Solution:

To enable an agent to learn the best actions in an unknown environment by interacting with it and using rewards, without needing a model of the environment. It helps in decision-making to maximize long-term rewards through trial and error.

Solution Statement:

We model the problem as a Markov Decision Process (MDP) and apply Q-learning to learn the optimal action in each traffic signal state.

Steps:

1. Initialize Q-table for all state–action pairs with zeros.

2. For each episode:

Observe current state (e.g., light color).

Choose an action using an exploration policy (e.g., **ε**-greedy).

Receive a reward and next state.

Update Q-value using the Q-learning equation.

Repeat until episode ends.

3. After enough training, extract the policy (best action per state) from the Q-table.

Example Q-Table (After Learning)

So, the optimal policy is:

Red → Stop

Yellow → Slow Down

Green → Go

State	Stop	Go	Slow Down	Best Action
Red	5	-20	1	Stop
Yellow	1	0	2	Slow Down
Green	-5	10	1	Go

Real-World Applications:

- Self-driving cars (traffic behavior)
- Game AI (learning strategies)
- Robotics (path planning)
- Finance (automated trading)
- Smart grids (energy management)