Date		Session No	7

Topic : BI-Unit 4 - Decision Making Techniques

Advanced Decision-Making Techniques: Models, Optimization, and Analytics

1. Decision Support Systems Modeling

Decision Support Systems (DSS) are interactive computer systems that help decision-makers use data, models, and technology to solve problems and make decisions. A DSS models the decision-making process and assists in choosing the best action by analyzing data.

Key Components of DSS:

- **Database**: Stores data needed for decision-making.
- **Model Base**: Contains mathematical models to analyze data.
- **User Interface**: Allows users to interact with the system and input their preferences.

---

2. Structure of Mathematical Models for Decision Support

In **decision support systems**, mathematical models help simulate different scenarios to aid decision-making. These models usually consist of:

- **Variables**: Represent factors that impact the decision.
- **Equations/Relationships**: Define how the variables interact.
- **Constraints**: Limitations on the decision (e.g., budget constraints).

- **Objective Function**: What we want to maximize or minimize (e.g., profit, cost).

Example:

A company may model a decision to minimize costs while meeting demand for products, with constraints like available resources and labor.

---

3. Certainty, Uncertainty, and Risk

- **Certainty**: The decision-maker knows all the outcomes and their probabilities. For example, buying a government bond with a fixed interest rate provides certainty of returns.
- **Uncertainty**: The decision-maker doesn't know all the outcomes or probabilities. For example, launching a new product without knowing how customers will respond.
- **Risk**: The decision-maker knows the probabilities of different outcomes but not the exact outcome. For example, investing in the stock market involves risk because you can estimate potential returns but can't predict them exactly.

---

4. Decision Modeling with Spreadsheets

Spreadsheets (like Excel) are often used for decision modeling because they allow you to input data, create models, and analyze different scenarios. Decision modeling involves:

- **Data Entry**: Input relevant data (e.g., sales, costs).
- **Formulas**: Use mathematical functions to calculate outcomes.
- **What-If Analysis**: Change values to see how different decisions affect outcomes.

Example (Excel):

1. Enter costs, sales, and profits in cells.
2. Use formulas like `=SUM()` to calculate totals.
3. Change inputs (e.g., increasing sales) and see the effect on total profit.

---

5. Mathematical Programming Optimization

Mathematical programming involves finding the best possible solution (optimum) to a problem, given certain constraints. The most common types are:

- **Linear Programming (LP)**: A method for optimizing a linear objective function, subject to linear constraints.
  Example: A factory wants to maximize profit by producing two products, subject to constraints like limited raw materials and labor hours.

Python Example (Using PuLP for Linear Programming):

```
from pulp import LpMaximize, LpProblem, LpVariable

# Define the problem
problem = LpProblem("Maximize_Profit", LpMaximize)

# Decision variables: number of products A and B
x = LpVariable("Product_A", lowBound=0)
y = LpVariable("Product_B", lowBound=0)

# Objective function: maximize profit
problem += 20 * x + 30 * y  # Profit from A and B

# Constraints
problem += 4 * x + 3 * y <= 240  # Raw material constraint
problem += 2 * x + 3 * y <= 180  # Labor hours constraint

# Solve the problem
problem.solve()

# Output the results
print(f"Produce {x.varValue} units of Product A and {y.varValue} units of Product B")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /home/sagar-v/.local/lib/python3.12/site-packages/pulp/solverdir/cbc/linux/64/cbc /tmp/3f436fd3a2914586916
14d89ccad2c39-pulp.mps -max -timeMode elapsed -branch -printingOptions all -solution /tmp/3f436fd3a291458691614d89ccad2c3
9-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 7 COLUMNS
At line 14 RHS
At line 17 BOUNDS
At line 18 ENDATA
Problem MODEL has 2 rows, 2 columns and 4 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve 2 (0) rows, 2 (0) columns and 4 (0) elements
0  Obj -0 Dual inf 50 (2)
0  Obj -0 Dual inf 50 (2)
1  Obj 1800
Optimal - objective value 1800
Optimal objective 1800 - 1 iterations time 0.002
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Produce 0.0 units of Product A and 60.0 units of Product B
```

6. Decision Analytics with Decision Tables and Decision Trees

- **Decision Tables**: Used to model decisions based on different conditions and their outcomes. Each row of the table represents a decision rule.

Example: A company decides on a discount rate based on customer type and order quantity.

Customer Type	**Order Quantity**	**Discount Rate**
New	< 50	5%
New	>= 50	10%
Returning	< 50	10%
Returning	>= 50	15%

Decision Trees: A graphical representation of possible decisions and their consequences. It helps map out the different choices and the probability of each outcome.

Example of a Decision Tree:

If a business is deciding whether to invest in a new product, the tree shows the possible scenarios (success, failure) and their associated probabilities.

Python Example (Using `sklearn` for Decision Tree Classification):

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris

# Load data
iris = load_iris()
X, y = iris.data, iris.target

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Create and train decision tree model
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Test the model
accuracy = model.score(X_test, y_test)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.8666666666666667
```

7. Multi-Criteria Decision Making with Pairwise Comparisons

In **Multi-Criteria Decision Making (MCDM)**, we often have to make decisions based on several conflicting criteria (e.g., cost vs. quality). **Pairwise comparisons** help rank the importance of these criteria by comparing them two at a time.

- **Example**: When buying a car, you might compare **price** against **fuel efficiency**, and **comfort** against **safety**.
  You can assign scores to each comparison, then aggregate the results to determine which criterion is most important overall.

Example Process:

1. Compare **Cost vs. Quality**: Is cost more important than quality?
2. Compare **Quality vs. Speed**: Is quality more important than speed?
3. Rank criteria based on the comparisons.

Pairwise comparisons are often used in techniques like **Analytic Hierarchy Process (AHP)** to find a weighted ranking of options.

Summary:

- **Decision Support Systems Modeling** helps businesses analyze data and make informed decisions.
- **Mathematical Models** use variables, equations, and constraints to simulate decision-making scenarios.
- **Certainty, Uncertainty, and Risk** define how much information the decision-maker has.

- **Decision Modeling with Spreadsheets** allows users to input data and test different scenarios using formulas.
- **Mathematical Programming Optimization** finds the best solution to a problem, considering constraints.
- **Decision Tables and Trees** help visualize decisions and their outcomes.
- **Multi-Criteria Decision Making** uses pairwise comparisons to prioritize criteria for decision-making.

END