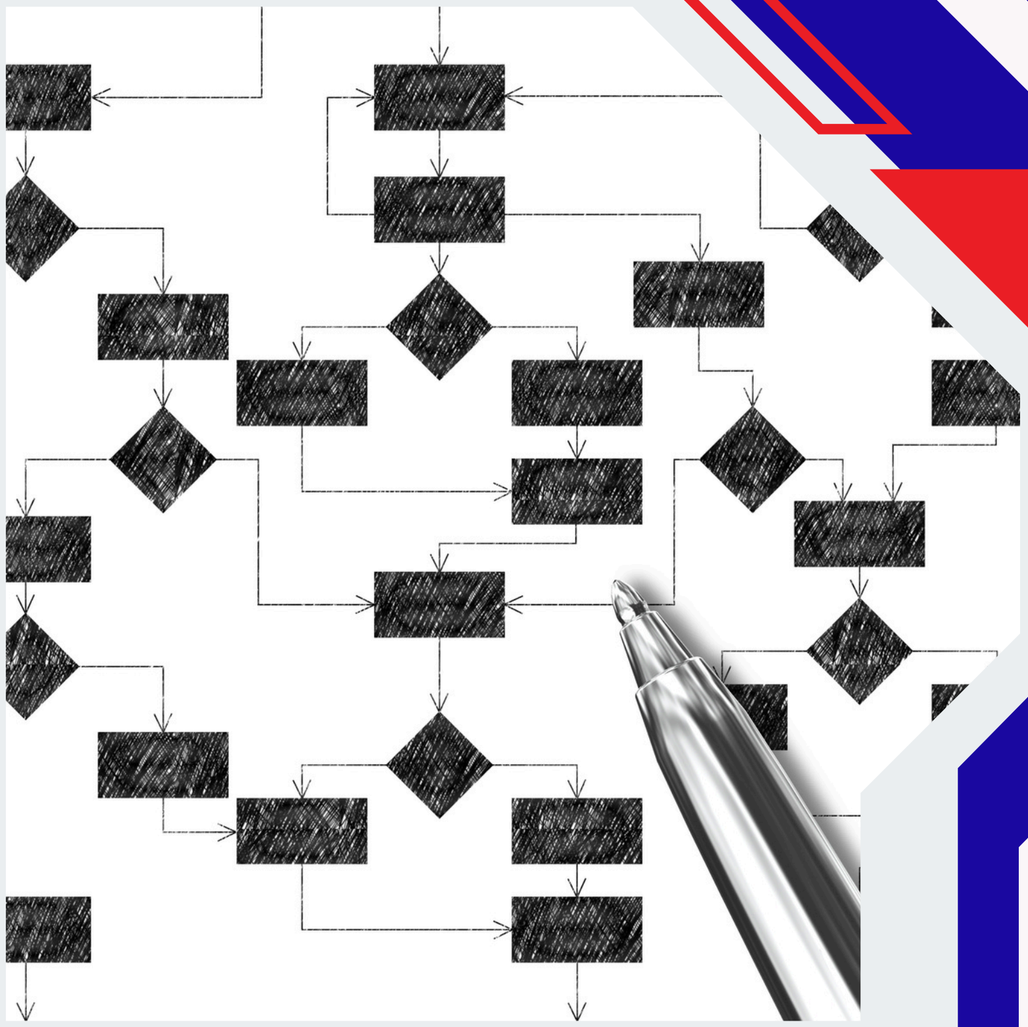
RULE-BASED AI: AN INTRODUCTION



INTRODUCTION TO AI | AI6310

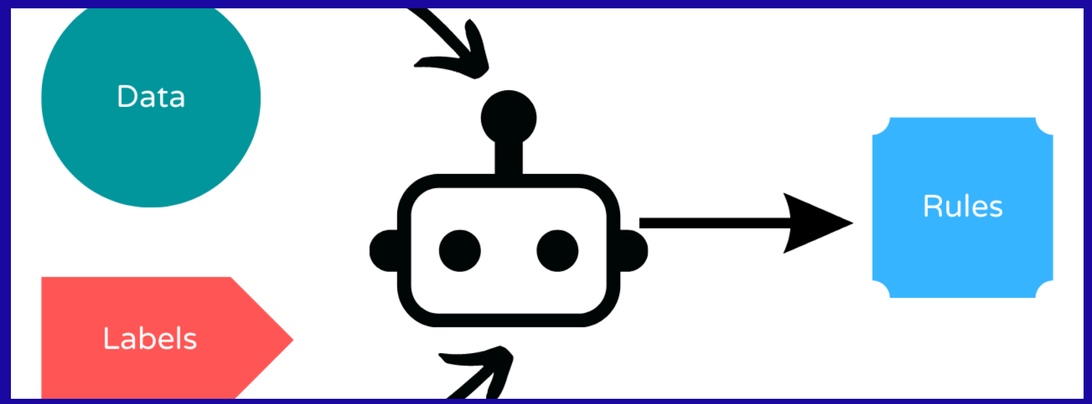
JOHN PAUL M. MANUEL | ACLC COLLEGE OF TAYTAY

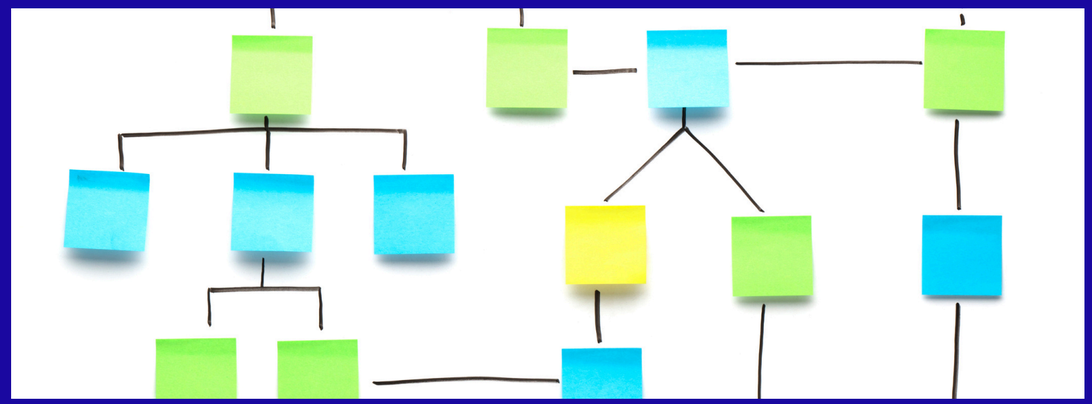
ACLC COLLEGE OF TAYTAY
(A Member of AMA Education System)

VERSION 01-250225-2425

WHAT IS RULE-BASED AI?

Rule-Based AI is a type of artificial intelligence that operates using a predefined set of rules and logical conditions to make decisions. These rules are typically written in if-then-else format, where an AI system follows a structured set of commands to process data and generate outputs.





RULE-BASED AI

HOW RULE-BASED AI WORKS

- Uses a knowledge base (a set of facts and rules).
- Employs an inference engine to apply rules to given inputs.
- The system matches conditions to execute predefined actions.
- Works like an expert system, where knowledge is explicitly programmed.

EXAMPLES OF RULE-BASED AI

- Chatbots (Early chatbots like ELIZA used pattern-matching rules).
- Spam Filters (Email services classify messages based on predefined rules).
- Medical Diagnosis Systems (Expert systems for decision support).
- Game AI (Chess engines with rule-based heuristics).
- Automated Customer Support (Rule-based decision trees for FAQs).

RULE-BASED AI

ADVANTAGES OF RULE-BASED AI

✓ Transparency & Interpretability – Easy to understand and debug.
✓ Predictability – Outputs are consistent and follow strict logic.
✓ Good for well-defined problems – Works well in structured environments.
✓ Low computational cost – Compared to data-driven AI, it requires less training power.

LIMITATIONS OF RULE-BASED AI

❌ Scalability Issues – Adding new rules increases complexity.
❌ Inability to Learn – Does not adapt to new data or improve over time.
❌ Limited in Unstructured Environments – Struggles with ambiguous or novel situations.
❌ Rule Conflicts – Overlapping rules can cause logical conflicts.

RULE-BASED AI

RULE-BASED AI VS MACHINE LEARNING

Feature	**Rule-Based AI**	**Machine Learning AI**
Learning Capability	No, rules are fixed	Yes, learns from data
Adaptability	Low	High
Transparency	High (explicit rules)	Often a "black box"
Best Use Cases	Well-defined problems	Dynamic, evolving tasks

RULE-BASED AI VS MACHINE LEARNING

RULE-BASED AI: EXPLICIT LOGIC AND PREDEFINED RULES

- Operates using a structured if-then-else set of rules.
- A human expert or developer defines all the rules explicitly.
- Uses a knowledge base (rules, facts) and an inference engine to apply rules to given inputs.
- Decisions are always based on predefined conditions.

Example:

A banking fraud detection system using Rule-Based AI:

- If a transaction exceeds $10,000 and originates from a new location, then flag it as suspicious.

RULE-BASED AI VS MACHINE LEARNING

MACHINE LEARNING AI: LEARNING FROM DATA

- ML systems analyze historical data to learn patterns and relationships instead of relying on hard-coded rules.
- Uses statistical models and algorithms like decision trees, neural networks, and deep learning.
- Instead of manual rule-setting, the model is trained on data and adjusts itself over time.

Example:

A banking fraud detection system using Machine Learning:

- Trained on millions of past transactions, the model learns patterns of fraud and identifies anomalies even when they don’t match predefined rules.

WHEN TO USE RULE-BASED AI VS. MACHINE LEARNING

✅ Use Rule-Based AI When:

✓ The problem is well-defined and follows a clear set of conditions.

✓ Transparency is important (e.g., compliance, auditing, medical diagnoses).

✓ There is limited or no historical data to train an ML model.

✓ Rules do not change often, and manual updates are manageable.

Examples:

- Spam filters using rule-based keyword detection.
- Access control systems (if user role = admin, then allow full access).
- Automated customer service scripts with pre-set responses.

WHEN TO USE RULE-BASED AI VS. MACHINE LEARNING

✅ Use Machine Learning When:

✔ The task involves large and complex datasets that would be impossible to handle with fixed rules.

✔ The system needs to adapt over time and learn new patterns.

✔ Accuracy improves with more data (e.g., personalized recommendations).

✔ The problem is too complex for manually defining rules.

Examples:

- Fraud detection in banking (ML models detect unusual patterns beyond predefined rules).
- Image recognition (ML learns to identify faces in photos).
- Speech recognition (ML understands spoken language by learning from voice data).
- Self-driving cars (ML adapts to real-world driving conditions).

RULE-BASED AI VS MACHINE LEARNING

CHALLENGES & LIMITATIONS

Challenges of Rule-Based AI:

- Not scalable: Managing thousands of rules becomes overwhelming.
- Hard to update: Requires manual intervention whenever new conditions arise.
- Fails with uncertainty: Cannot handle situations that are not predefined.

Challenges of Machine Learning AI:

- Requires large datasets: Needs vast amounts of labeled data to train effectively.
- Black-box nature: Hard to explain why a model made a specific decision.
- Computationally expensive: Requires significant processing power, especially for deep learning.

RULE-BASED AI: CHATBOT AND REGEX

WHAT IS REGEX?

- Regular Expressions (Regex) are patterns used to match character combinations in strings.
- They are powerful for searching, replacing, and validating text.

Creating a Regex in JavaScript

- Using literal syntax: /pattern/flags
- Example: let regex = /hello/i;

RULE-BASED AI: CHATBOT AND REGEX

COMMON REGEX FLAGS

- i – Case insensitive
- g – Global search
- m – Multi-line search
- u – Unicode support
- s – Allows . to match newline characters

BASIC REGEX PATTERNS

- ^ – Matches the start of a string
- $ – Matches the end of a string
- . – Matches any character (except newline)
- * – Matches 0 or more occurrences
- + – Matches 1 or more occurrences
- ? – Matches 0 or 1 occurrence

CHARACTER CLASSES

- \d – Matches any digit (0-9)
- \w – Matches any word character (a-z, A-Z, 0-9, _)
- \s – Matches any whitespace character
- \D – Matches any non-digit
- \W – Matches any non-word character

RULE-BASED AI: CHATBOT AND REGEX

REGEX EXAMPLES

- Validate an email: /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/
- Validate a phone number: /^\d{10}$/
- Extract numbers from a string: /\d+/g
- Example: "Order #1234 and #5678".match(/\d+/g) // ["1234", "5678"]
- Match a URL: /https?:\/\/(www\.)?[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}/
- Match a date format (YYYY-MM-DD): /\d{4}-\d{2}-\d{2}/
- Match a hexadecimal color code: /#([a-fA-F0-9]{6}|[a-fA-F0-9]{3})/