Module 2 · Session 2

Generic problems and applications, core AI/ML techniques/algorithms

The AI Spectrum — From Classical AI to Agentic AI

Leadership with AI - IIT Bombay, Rajendra M Sonar, Ph.D.

What we are going to learn

Understanding what core problems AI/ML solves, core techniques and how they are modelled and used

1. Generic problem types
Classification, clustering, regression, intelligent matching, recommender systems

2. Generic applications
Forecasting, sentiment analysis, segmentation, market-basket analysis, etc.

Understanding Semantic and Core AI

3. Hybrid systems
Integrating techniques to solve the problems in better way.

4. Knowledge-based systems
How systems like expert systems play role in managing knowledge required for problem solving

What are the business problems?

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

ML/AI becomes valuable when important decisions depend on predicting uncertain outcomes from data

Telecom
What is the chance a customer will churn?

Banking
Will this borrower repay the loan?

Insurance
Is this claim likely to be fraudulent?

Retail / E-commerce
Which product is this customer most likely to buy?

Why ML/AI?

Use data to estimate likelihoods, scores, patterns, and recommendations that support better decisions.

Manufacturing
Which machine is likely to fail next?

Healthcare
Which patient is at risk of readmission?

HR / People
Which employee is most likely to leave?

Supply Chain
What demand should we expect next period?

Business data → ML/AI model → Prediction / score / recommendation → Better business decision

In many industries, the core decision is not only 'what happened?' but 'what is likely to happen next, and what should we do about it?'

Typical generic problem types*

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

AI Problem Solving

turning data into decisions

1 Classification
- Assign to a category
- Approve / reject
- Fraud / no fraud

Used when the goal is to label or categorize.

2 Clustering
- Find natural groups
- Customer segments
- Behavior patterns

Used when the goal is to discover hidden structure.

3 Regression / Prediction
- Estimate a value
- Forecast demand or sales
- Predict risk or cost

Used when the goal is to predict a number or future outcome.

4 Optimization
- Choose the best option
- Scheduling and routing
- Resource allocation

Used when the goal is to find the best action under constraints.

5 Recommendation
- Suggest next best action
- Recommend product or content
- Rank alternatives

Used when the goal is to personalize and prioritize choices.

6 Generation / Synthesis
- Create content
- Summarize or draft
- Generate plans or ideas

Used when the goal is to create useful outputs from information.

Together, these categories help organizations analyze, predict, decide, and act.

This provides an intuitive bridge from human-like intelligence to AI-based problem solving.

Example: from problem to action: Will borrower repay the loan?

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Business User	ML Developer				Business User
1. Business Problem	2. Data Required	3. Core Problem Type	4. Possible ML Algorithm	5. Results	6. Interpretation for Business Decision
"Will this borrower repay the loan?"	• Credit score • Income and employment stability • Existing loans / debt-to-income ratio • Past repayment history • Loan amount and tenure	Binary Classification • Class 1: likely to repay • Class 2: likely to default • Predict probability of repayment/default	Logistic Regression • Simple and interpretable • Good baseline for credit risk • Can output probability score Other options: Decision Tree, Random Forest, XGBoost	Predicted probability of repayment = 0.87 Predicted probability of default = 0.13 • Model classifies borrower as Low Risk • Example model performance: Accuracy 84%, AUC 0.89	• High repayment probability suggests approval • Can set loan amount / pricing / collateral based on risk • If probability were low, reject or ask for safeguards Decision: Approve loan (possibly at standard terms)

This is how ML turns a business decision into a structured problem: question → data → prediction → interpretation → action.

Example: Business questions in marketing

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

A simple way to explain how business decisions translate into AI problem types

Business Question Type	ML/AI Category
Who should we target?	Classification / Scoring
Which customers are similar?	Clustering
What will happen next?	Prediction / Forecasting
What should we recommend?	Recommendation
Which action gives best return?	Optimization
What are customers saying?	NLP / Sentiment / Topic Modeling
What message/content should we create?	Generative AI
What caused the conversion?	Attribution / Causal AI

AI becomes easier to understand when business questions are translated into problem types.
This helps executives connect real decisions with the right ML/AI approach.

Typical generic problem types*

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

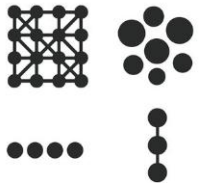
Apply

Think, plan & reason

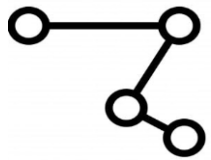
Problem → Solution

Each of these generic problem can address wide range of applications.

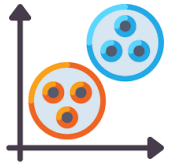
Regression/
Time-series



Pattern Recognition



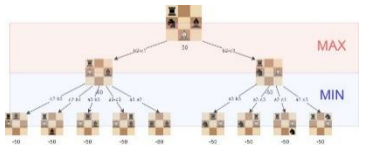
Association Rule Mining



Clustering

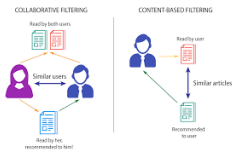
Classification

Pattern Matching



Game Playing

Diagnostics



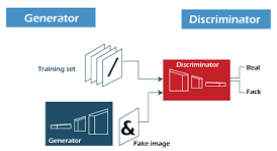
Recommendation

Optimization

Search/
Information retrieval

Summarization

Self learning



Content generation

Entity Matching
(including semantic)

Descriptive Analytics

Predictive | Descriptive | Prescriptive | Customization flexibility | Problem Complexity | Operationalize | Contextualize | Interconnect

* Please note some of the problem types/techniques were conventionally part of data mining but converging into ML kind of problem solving.

Typical generic application areas

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

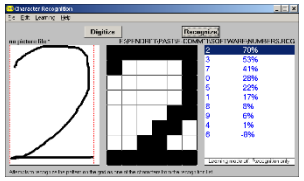
Apply

Think, plan & reason

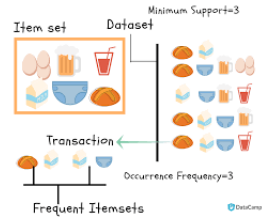
Problem → Solution

Generic application areas can be based on: domain, data type/ format, entity type (product, content…), etc.

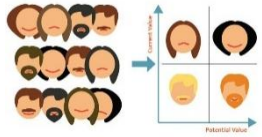
Prediction



OCR (image to text)

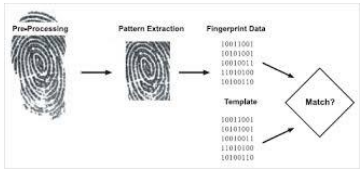


Associating products (market basket analysis)

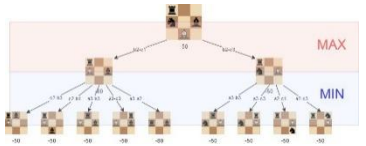


Customer Segmentation

Sentiment analysis



Fingerprint Matching



Chess game

Automated help desk

Content P&R

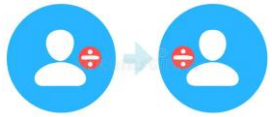
Route optimization

Product Search

Content Insights

Game playing

Text generation



Profile matching

Rating Statistics

Applications

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Each generic problem can be mapped to multiple practical application

Sales forecasting

KYC data extraction

Ecom: People bought together

Segment based marketing

Feedback Analysis

Fingerprint Authentication

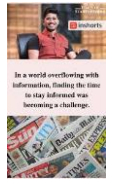
Chess game

Automated Ticketing

News recommendation

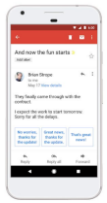
Delivery optimization

Personalized search



News Insights

Chess Game



Email response generation

Match-making

Customer Analysis

Business Problem → App → Generic Application → Generic Problem Type

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

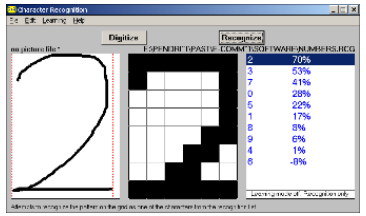
Solution

Each generic problem can be mapped to multiple practical application

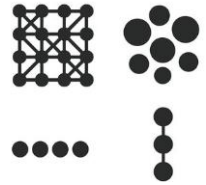
Business Problem

KYC App

KYC data extraction



OCR (image to text)



Pattern Recognition

Examples

Example: Character recognition

Generic Application	• Character recognition: Optical character recognition
Core Learning Type	• Data driven • Supervised learning
Problem Type	• Pattern recognition • Image pattern recognition • Character recognition • Machine Learning
Technique/ Technology	• Artificial neural network • Architecture: Multilayer perceptron
Data	• Image: scanned hand-written characters • Supervised learning: characters needs to be labelled A,B,…Z, 0,2,3,…9
Feature Extraction	• Convert images into to black and white, edge detection - boxing, converting into binary numbers
Implementation	• Matlab

Example: Recommendation

	• Recommendations in eCommerce app/website
	• Data driven • Supervised learning
	• Association mining/Collaborative filtering
	• Apriori Algorithm
	• Click-stream data (user id, item bought, rating, etc.)
	• R

Generic tasks amongst entities

Acquire → Learning

Memory
Adaptability

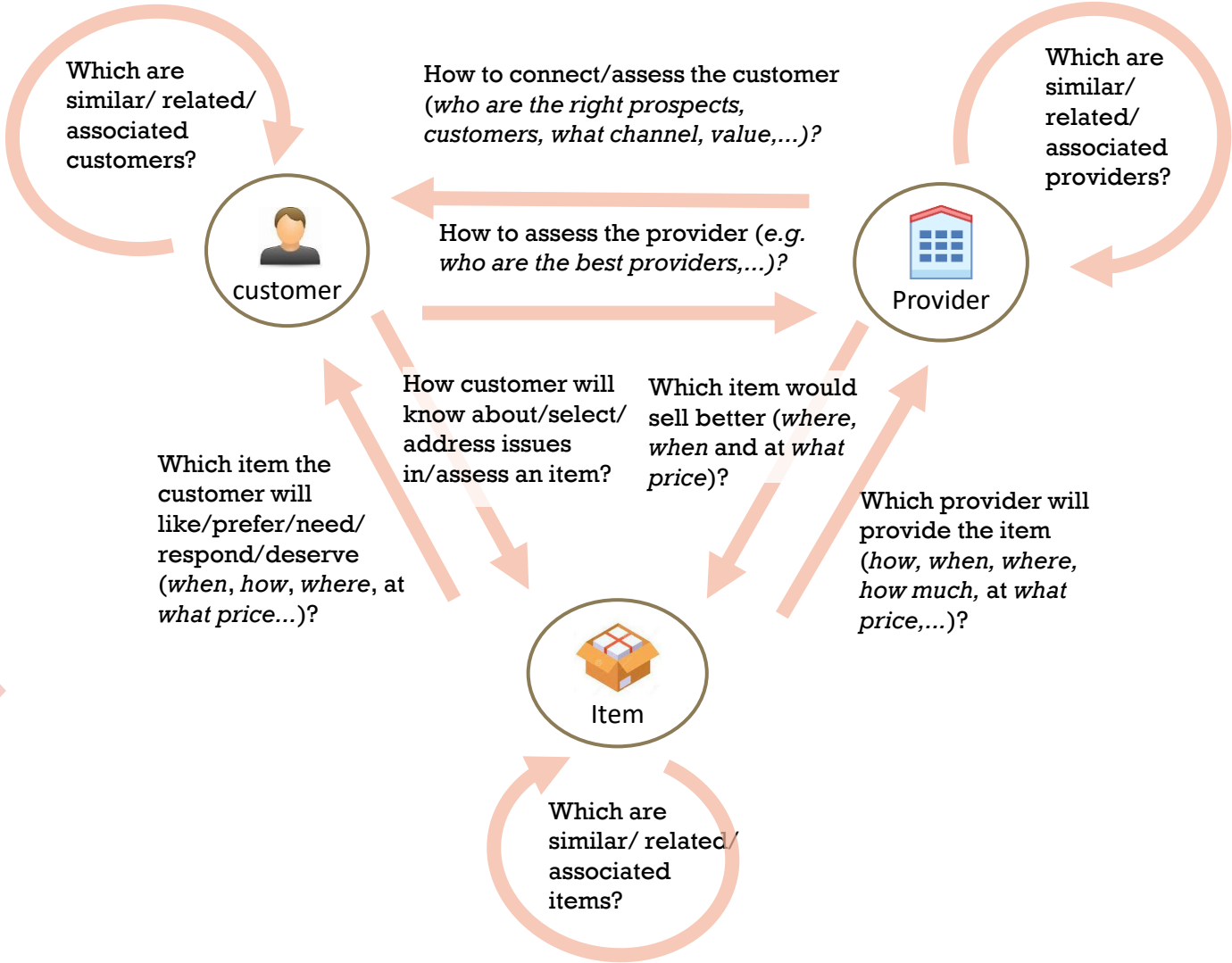
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution



Generic tasks amongst entities

Acquire → Learning

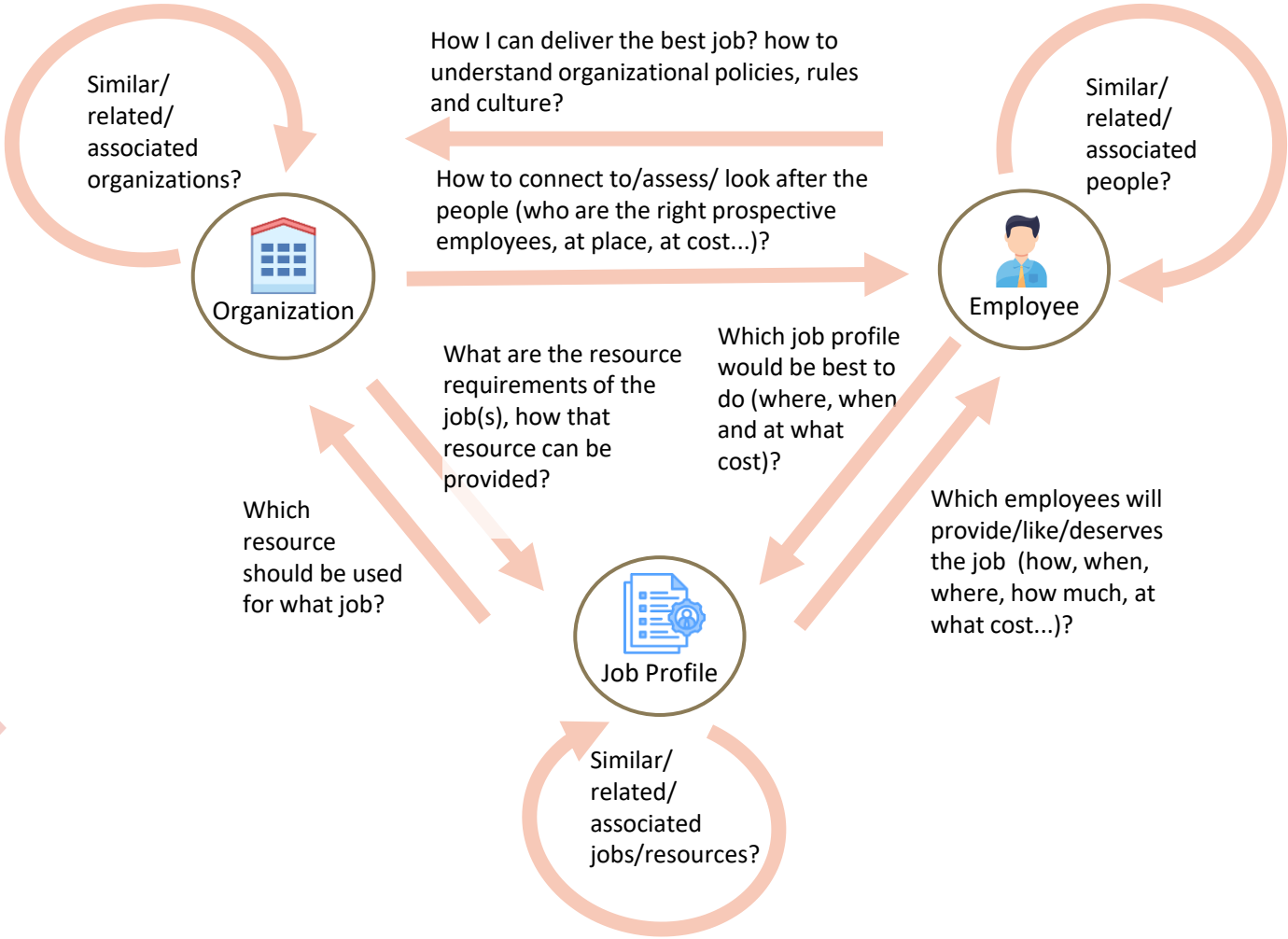
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Generic tasks amongst entities

Acquire → Learning

Memory
Adaptability

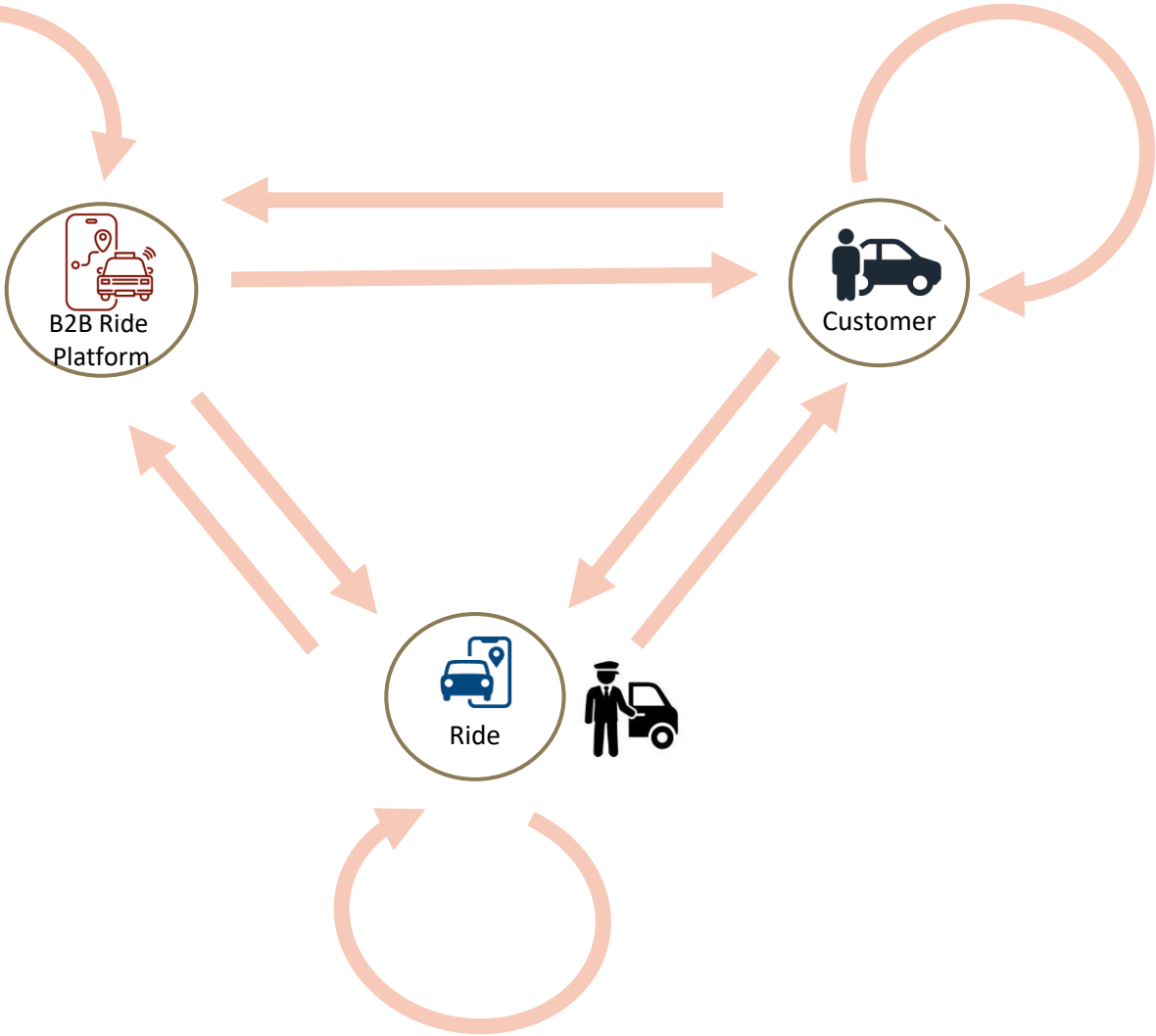
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution



Example Algo: Intelligent Matching: Possible generic Use-cases

Intelligent matching algorithm itself can address various use-cases including hyper personalization and recommendation

Acquire → Learning

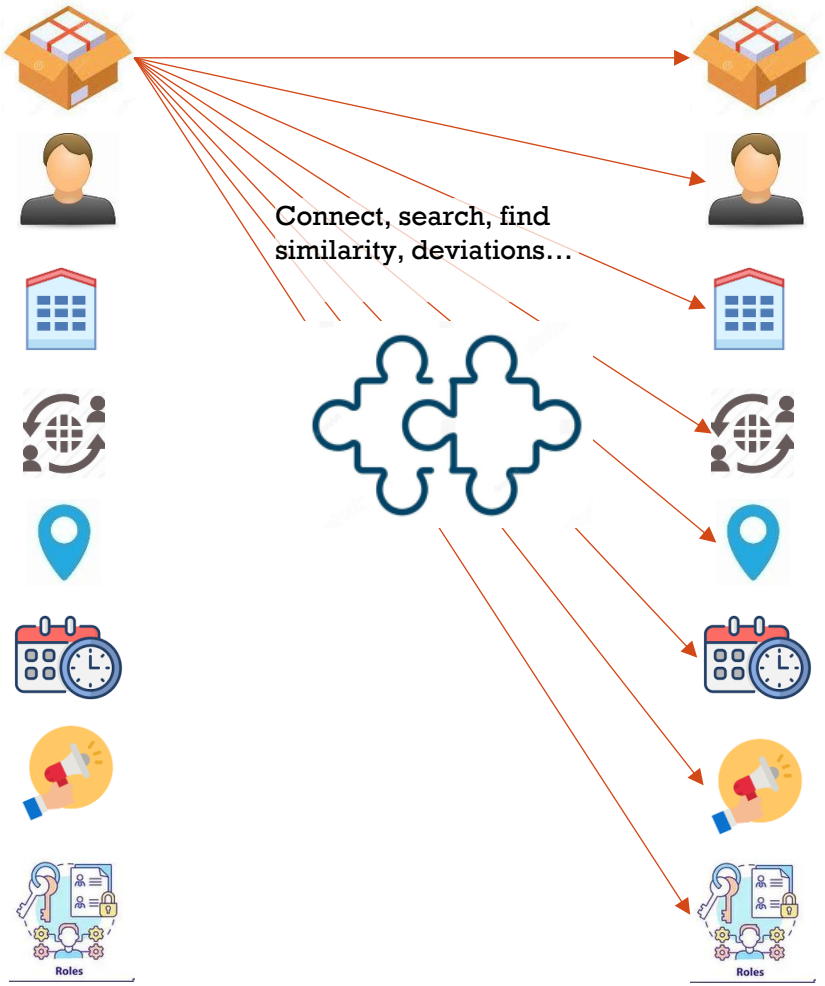
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Imagine you have detailed data and metadata about all your entities and domain knowledge and intelligent algos to connect them! *Everything will be done rightly* (at right time, location, cost, resource, product, stakeholder etc.). Entity may be just specification, requirement etc. For example, item requirement -> item, searching product based on product requirement.

Paired

Similar

Part of

…

Example Algo: Intelligent Matching: Possible generic Use-cases

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

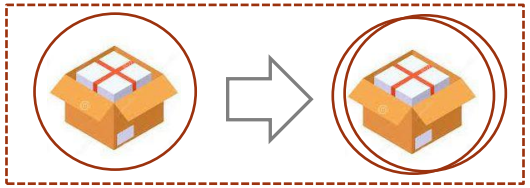
Apply

Think, plan & reason

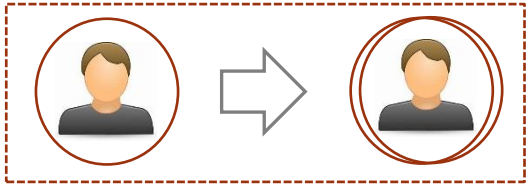
Problem

Solution

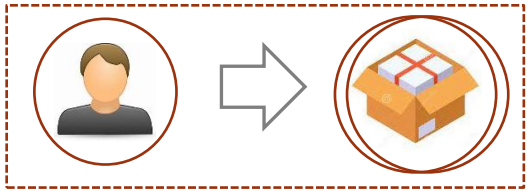
Matching entities



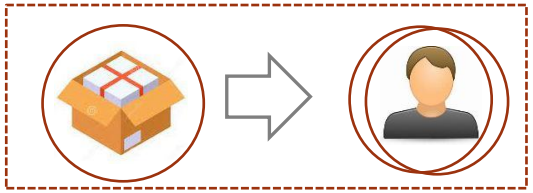
Similar kind of items …



Similar users, understand likely behaviour and risk…



Recommending/cross-selling/up-selling right product to the user



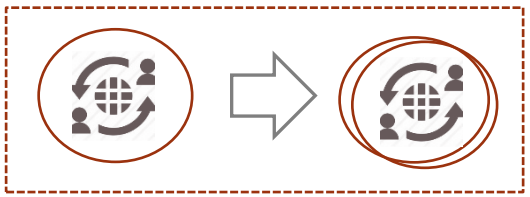
Right target user for selected/new item

Example

YouTube	Recommends video based on currently selected video
LinkedIn	Send invitation link to connect people you make like.
NETFLIX	Recommends contents based on what user has liked/seen in the past
LinkedIn	Sends emails to right people for a job profile

Example

Channels | Touch points



Transaction monitoring: Similar transactions, deviations from transactions done!

Example Algo: Intelligent Matching: Possible generic Use-cases

Intelligent matching generic use-cases

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Lot of insights can be drawn from matching entities. At the same time if no matching entities found means deviations from source entity.

Entity to be analyzed e.g. new customer, new transaction etc.

Source entity → Source entity profile

Profiles of stored entities ← Stored entities

Cluster matching profiles → Segment profiles → New entities

Creating new entities for N=G (e.g. Product/bundled offer catering to specific needs of clustered profile (e.g. customers with common behaviour)

Source entity profile + Profiles of stored entities → AI Context Matching Engine + Domain Model

Can be used to determine **how similar** source entity is to other (e.g. comparing two job profiles)

Similarities

Deviations

Deviation detection: to determine how much source entity is deviating from the rest ones (e.g. how much new transaction is deviating from past ones).

Can help to know **what is likely to happen** next e.g. next event in the sequence

What is next? | What is likely behaviour?

What is likely behaviour of source entity (e.g. whether subscriber is likely churn out or borrower is likely to default?)

Intelligent reuse

Most matching entities

Connecting the source entity to the right available entity(ies)

Pulling or pushing entities that match with source entity based on *model* and *profile*: past behaviour, requirement, relevance, utility, preference, demographics, cross-sell and up-sell opportunity, etc.

Optimize entities

New entity

Creating new entity N=1 (e.g. product, bundled or **customized offer** that suits to source entity)

Example Algo: Intelligent Matching: Possible generic Use-cases

Example use cases

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

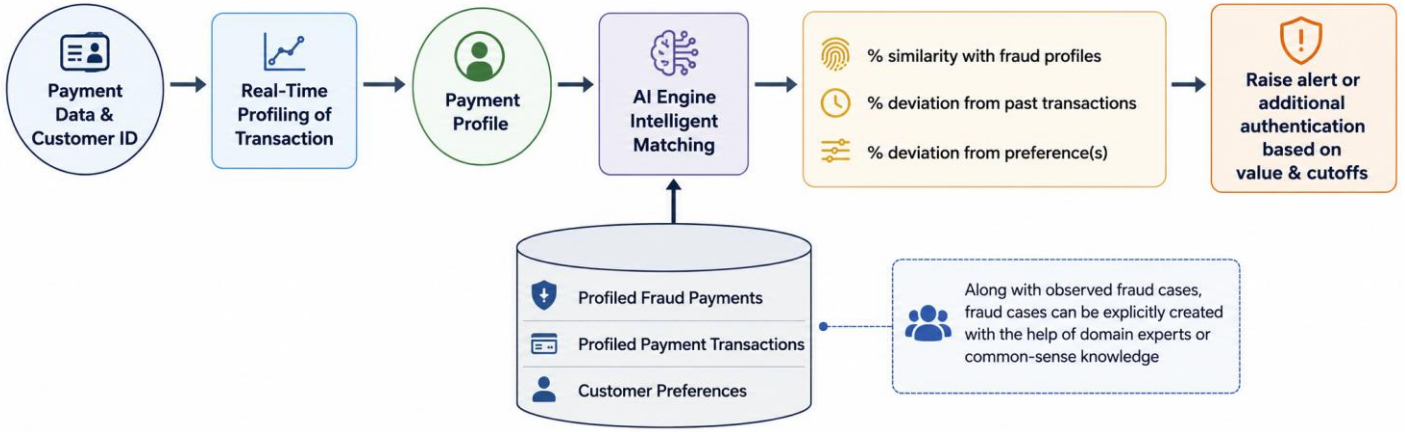
Apply

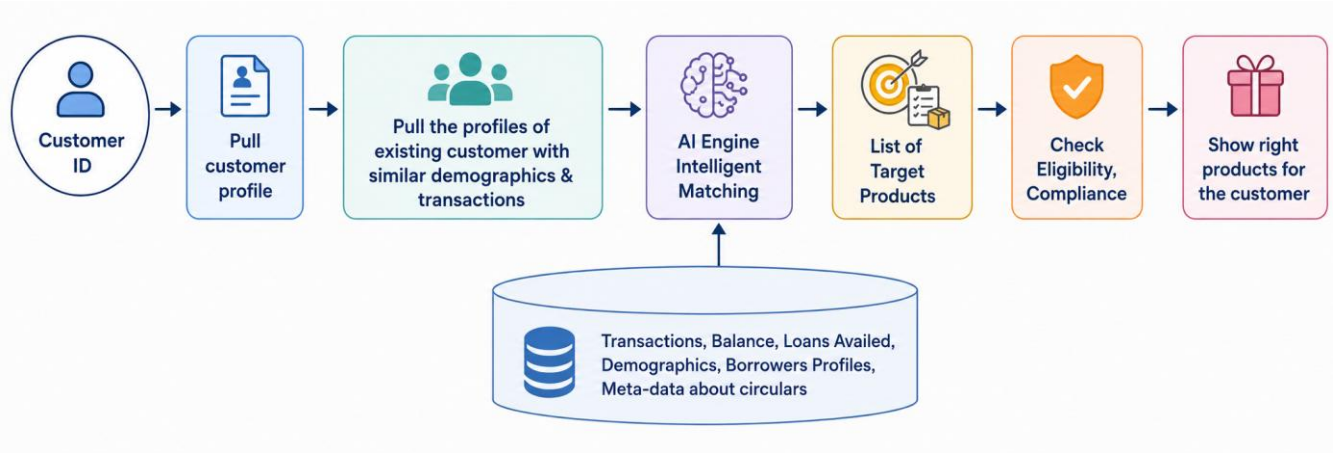
Problem

Think, plan & reason

Solution

Detect suspicious payment transaction using intelligent matching approach





NLP specific problems

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

NLP has issues and limitations especially when implementing in-house. Know NLP does not include computational capabilities so better model numeric parameters separately.

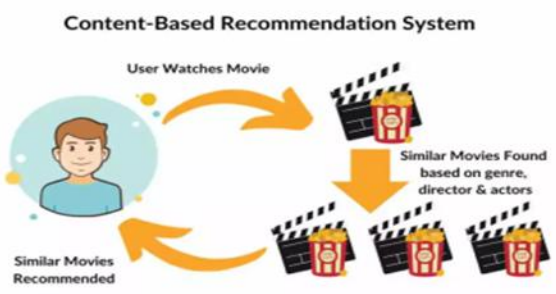
Automatic Summarization
Intelligently shortening long pieces of text

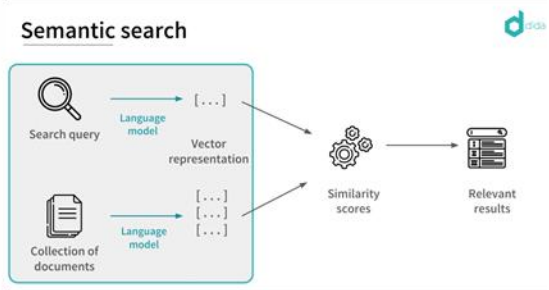
Named entity recognition
Locate and classify named entities pre-defined categories such as the organizations; person names; locations etc.

Sentiment analysis
To identify, for instance, positive, negative and neutral opinion form text or speech widely used to gain insights from social media comments, forums or survey responses

Speech recognition
Enables computers to recognize and transform spoken language into text – dictation – and, if programmed, act upon that recognition – e.g. in case of assistants like Google Assistant Cortana or Apple's Siri

Topic segmentation
Automatically divides written texts, speech or recordings into shorter, topically coherent segments and is used in improving information retrieval or speech recognition





Semantic Search

Semantic Step by step guide to NLP

Understanding Classification

Acquire → Learning

Memory
Adaptability

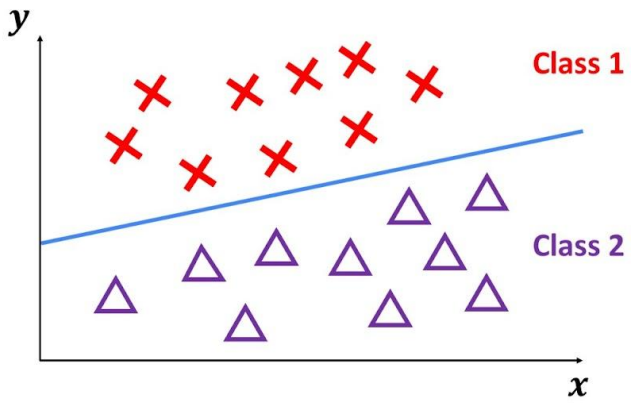
Tacit | Explicit | Implicit

Apply

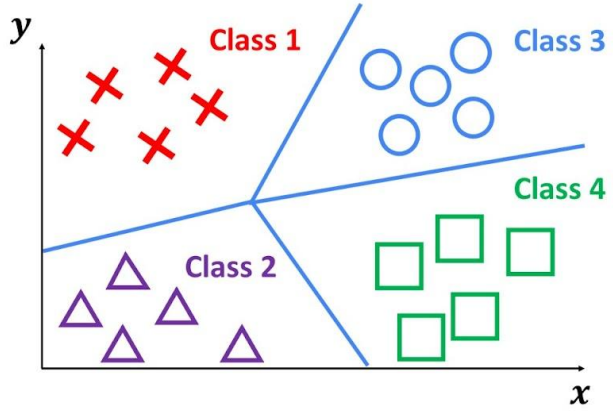
Think, plan & reason

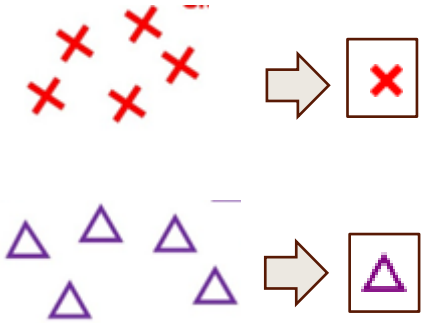
Problem → Solution

Binary Classification



Multiclass Classification





Confusion Matrix

		Predicted	
		✕	△
Actual	✕	✕	△
	△	✕	△

Correctly Predicted

Incorrectly Predicted

Understanding Classification

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Domain	Input	Output Class
Banking	Customer profile	Loan approved / rejected
Healthcare	Patient symptoms	Disease / no disease
Education	Student performance	Pass / fail
Marketing	Customer behavior	Will buy / will not buy
Cybersecurity	Network activity	Normal / suspicious

Feature	Meaning	Type
Age	Age of applicant	Numeric
MonthlyIncome	Monthly income of applicant	Numeric
ExistingLoans	Number of existing loans	Numeric
CreditScore	Creditworthiness score	Numeric
LoanApproved	Whether loan is approved or not	Binary class

Age	Monthly Income	Existing Loans	Credit Score	Loan Approved
25	25000	1	650	0
30	40000	2	700	1
22	18000	0	620	0
45	80000	1	760	1
35	55000	3	720	1
28	30000	2	640	0
50	90000	0	780	1
40	60000	2	710	1
23	22000	1	600	0
38	45000	4	680	0
48	75000	1	740	1
33	35000	3	660	0

Here, `LoanApproved` is the class label.
`1` means loan approved.
`0` means loan not approved.

Understanding Classification

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

```python
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

data = [
    [25, 25000, 1, 650, 0],
    [30, 40000, 2, 700, 1],
    [22, 18000, 0, 620, 0],
    [45, 80000, 1, 760, 1],
    [35, 55000, 3, 720, 1],
    [28, 30000, 2, 640, 0],
    [50, 90000, 0, 780, 1],
    [40, 60000, 2, 710, 1],
    [23, 22000, 1, 600, 0],
    [38, 45000, 4, 680, 0],
    [48, 75000, 1, 740, 1],
    [33, 35000, 3, 660, 0],
    [29, 50000, 1, 705, 1],
    [55, 95000, 2, 790, 1],
    [26, 28000, 3, 630, 0],
    [42, 70000, 2, 735, 1],
    [31, 32000, 0, 675, 0],
    [36, 58000, 1, 715, 1],
    [24, 26000, 2, 610, 0],
    [46, 85000, 3, 755, 1]
]

df = pd.DataFrame(data, columns=[
    "Age",
    "MonthlyIncome",
    "ExistingLoans",
    "CreditScore",
    "LoanApproved"
])
```

```python
X = df[["Age", "MonthlyIncome", "ExistingLoans", "CreditScore"]]
y = df["LoanApproved"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=7, stratify=y
)

model = DecisionTreeClassifier(max_depth=3, random_state=7)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("Test Data with Prediction")
result = X_test.copy()
result["Actual"] = y_test
result["Predicted"] = y_pred
print(result)

print("\nAccuracy:", accuracy_score(y_test, y_pred))

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

plt.figure(figsize=(12, 6))
plot_tree(
    model,
    feature_names=X.columns,
    class_names=["Not Approved", "Approved"],
    filled=True
)
plt.show()
```

Understanding Classification

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Console

Test Data with Prediction

	Age	MonthlyIncome	ExistingLoans	CreditScore	Actual	Predicted
17	36	58000	1	715	1	1
15	42	70000	2	735	1	1
13	55	95000	2	790	1	1
2	22	18000	0	620	0	0
14	26	28000	3	630	0	0
9	38	45000	4	680	0	1

Accuracy:

0.8333333333333334

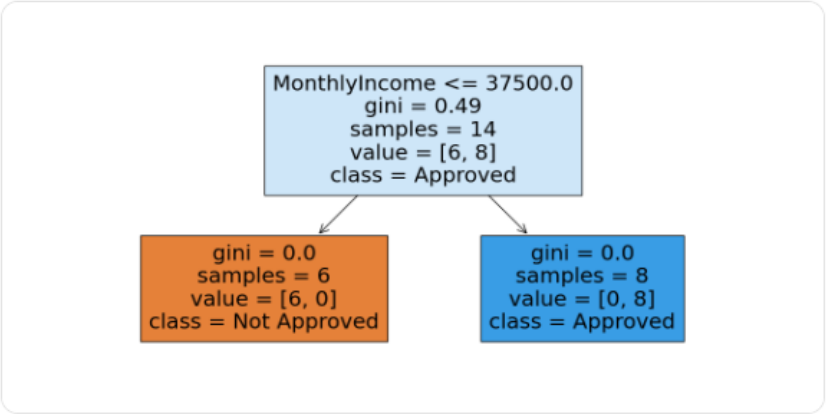
Confusion Matrix:

```
[[2 1]
 [0 3]]
```

		Predicted	
		Not Approved	Approved
Actual	Not Approved	2	1
	Approved	0	3

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.67	0.80	3
1	0.75	1.00	0.86	3
accuracy			0.83	6
macro avg	0.88	0.83	0.83	6
weighted avg	0.88	0.83	0.83	6



Run completed in 11538.899999856949ms

Applicants with higher income and better credit score are more likely to be approved. Applicants with lower income, lower credit score, or too many existing loans are less likely to be approved.

3 approved applicants were correctly predicted as approved. 2 not-approved applicants were correctly predicted as not approved. 1 not-approved applicant was wrongly predicted as approved.

Input:
Age = 36
MonthlyIncome = 58000
ExistingLoans = 1
CreditScore = 715

Output:
LoanApproved = 1

Understanding Classification

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

	Not Approved	Approved
Not Approved	2	1
Approved	0	3

Precision
Precision answers:
Out of all cases **predicted** as this class, how many were actually correct?

For Class 0: Not Approved
`precision = 1.00`
This means: Whenever the model predicted `Not Approved`, it was always correct.
So, no approved customer was wrongly rejected.

For Class 1: Approved
`precision = 0.75`
This means: Out of all customers predicted as `Approved`, 75% were actually approved.
So, one customer was wrongly classified as approved.

Accuracy
`accuracy = 0.83`
This means: The model correctly predicted 5 out of 6 test records. So:
`Accuracy = 5 / 6 = 0.83`
or approximately **83% accuracy**.

	Not Approved	Approved
Not Approved	2	1
Approved	0	3

Recall
Recall answers:
Out of all **actual cases** of this class, how many did the model correctly identify?

For Class 0: Not Approved
`recall = 0.67`
This means: Out of 3 actual not-approved customers, the model correctly identified only 2.
So, 1 not-approved customer was wrongly predicted as approved.

For Class 1: Approved
`recall = 1.00`
This means: Out of 3 actual approved customers, the model correctly identified all 3.
So, the model did not miss any truly approved customer.

F1-Score
F1-score is a combined measure of precision and recall.
F1-score is useful when we want to balance both precision & recall.

Class	F1-score	Meaning
0	0.80	Good performance for not-approved class
1	0.86	Slightly better performance for approved class

Understanding Classification

Acquire → Learning

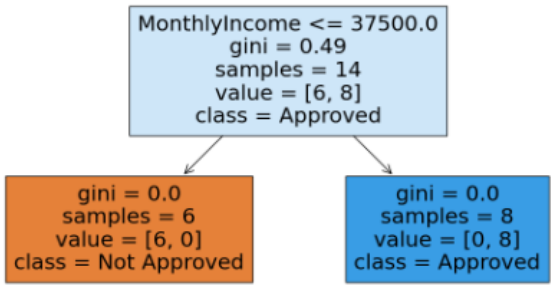
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Gini impurity tells us how mixed the classes are in a node. At the root node, there are 6 Not Approved and 8 Approved records, so the node is highly mixed and has gini 0.49. After splitting on MonthlyIncome, the two child nodes become pure, with gini 0.0. This means the split has successfully separated the classes.

```
Run completed in 11538.899999856949ms
```

Item	Meaning
`MonthlyIncome <= 37500.0`	Decision rule used for splitting
`gini = 0.49`	Impurity/mixing of classes
`samples = 14`	14 training records reached this node
`value = [6, 8]`	6 records are class 0, 8 records are class 1
`class = Approved`	Majority class is Approved

Understanding Classification

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

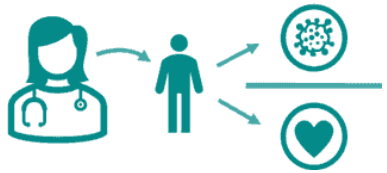
AUC (Area Under the ROC Curve, ROC: Receiver Operating Characteristic): In classification, accuracy tells us how many final predictions are correct at one selected threshold. AUC tells us how well the model separates the two classes across all possible thresholds. AUC is especially useful when we care about ranking customers by probability, such as ranking borrowers by default risk, customers by churn risk, or transactions by fraud risk

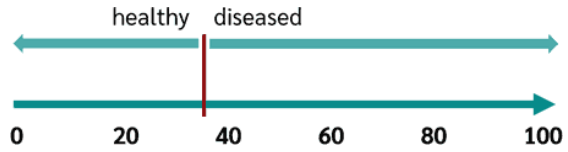
AUC tells us how many predictions were correct? whether the model gives higher scores to actual positive cases than to actual negative cases.

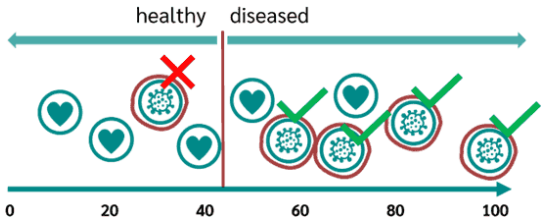
The ROC curve compares:

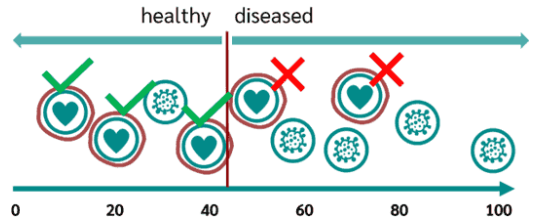
Metric	Meaning
True Positive Rate	How many actual positives were correctly detected
False Positive Rate	How many actual negatives were wrongly classified as positive

AUC Value	Interpretation
0.50	Model is no better than random guessing
0.60–0.70	Weak model
0.70–0.80	Fair model
0.80–0.90	Good model
0.90–1.00	Very strong model
1.00	Perfect separation









The **True Positive Rate** is plotted on the **y-axis**

and the **False Positive Rate** on the **x-axis**.

[ROC Curve and AUC Value] [ROC Curve: A Beginner's Guide]

Understanding Classification

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Problem

Think, plan & reason

Solution

Type I and Type II errors

Confusion Matrix (health context)

		Predicted	
		Positive	Negative
Actual	Positive	Correctly Predicted (True)	Incorrectly Predicted (False)
	Negative	Incorrectly Predicted (False)	Correctly Predicted (True)

Positive (patients with good health)

Negative (patients with poor health)

Correctly Predicted (True)

Incorrectly Predicted (False)

Type I Error — Healthy but predicted as unhealthy (false positive)

Type II Error — Unhealthy but predicted as healthy (false negative)

Understanding Classification: Results

Acquire → Learning

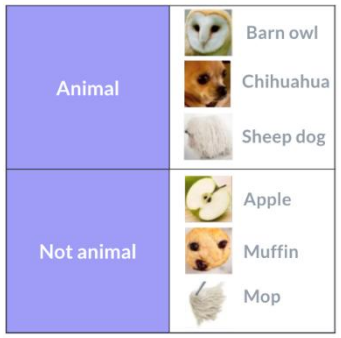
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Positive

Negative

+ True: means predicted correctly
- False: means predicted incorrectly

Actual \ Predicted	Animal	Not animal
Animal	True Positives (+)	False Negatives (-)
Not animal	False Positives (+)	True Negatives (-)

Confusion Matrix

ACTUAL \ PREDICTED	APPLE	GRAPES	BANANA	ORANGE
APPLE	10	2	1	2
GRAPES	5	12	1	2
BANANA	5	2	18	10
ORANGE	11	3	1	15

What is Accuracy, Precision, Recall and F1 Score?

Understanding Classification: Results

Acquire → Learning

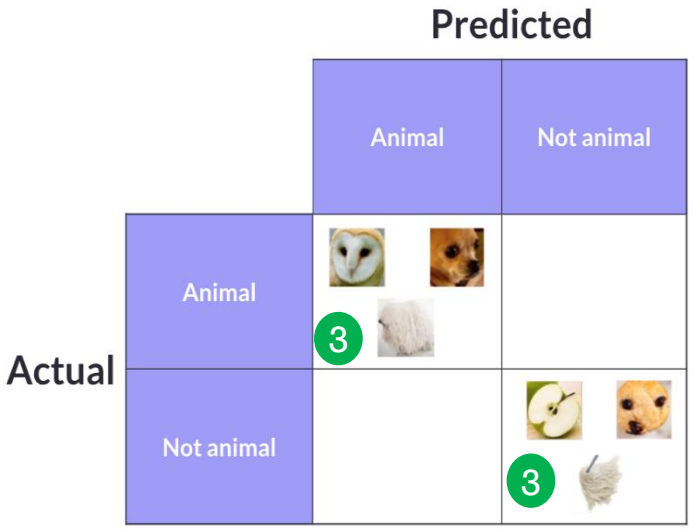
Memory
Adaptability

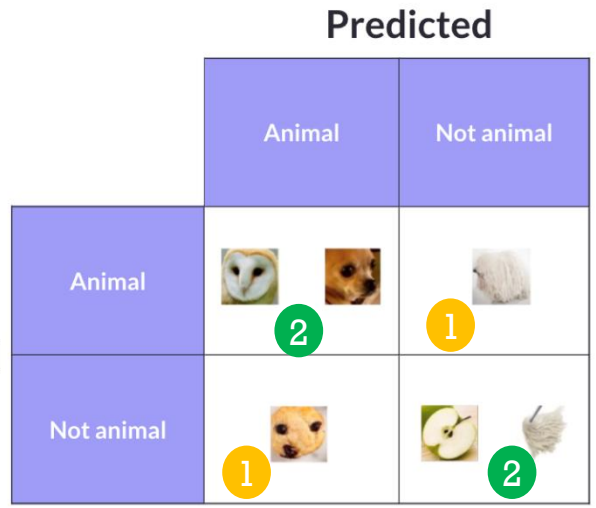
Tacit | Explicit | Implicit

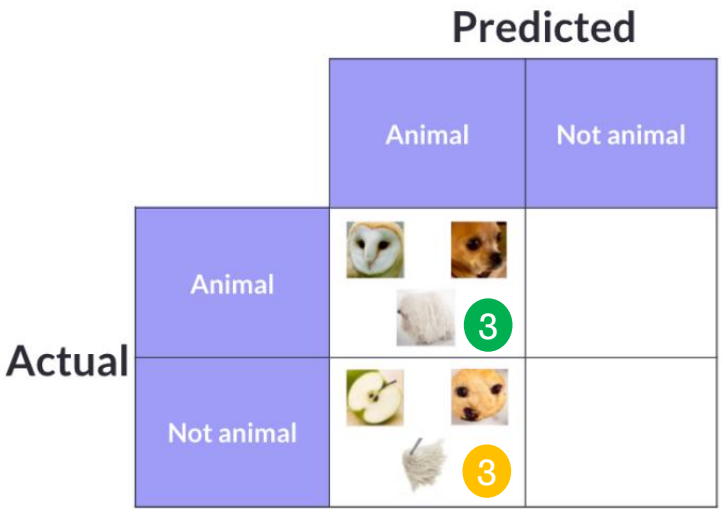
Apply

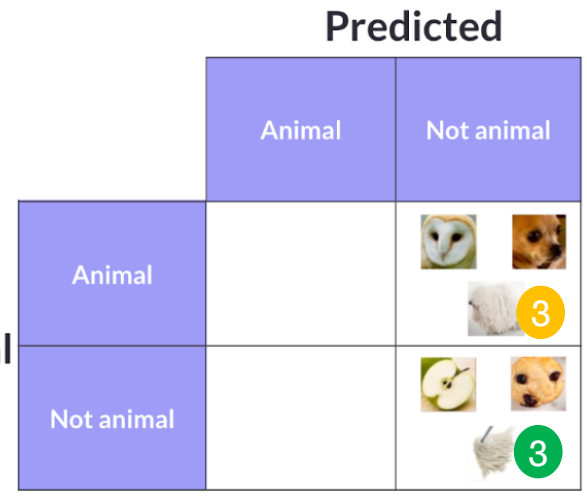
Think, plan & reason

Problem → Solution









[What is Accuracy, Precision, Recall and F1 Score?]

Understanding Classification: Results

Acquire → Learning

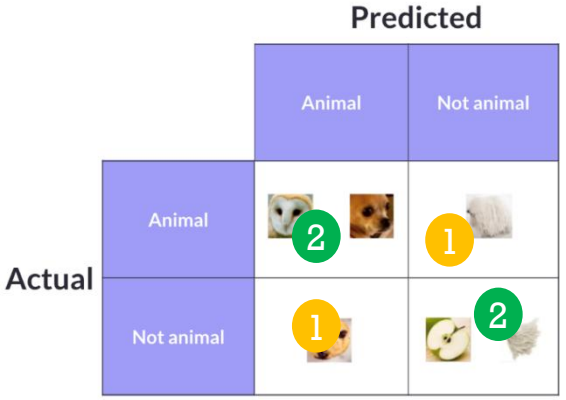
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



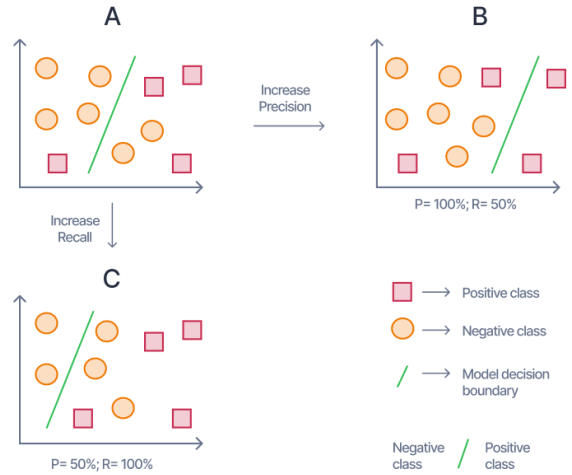
Accuracy is the most common metric to be used in everyday talk. Accuracy answers the question **"Out of all the predictions we made, how many were true?"**

Precision is a metric that gives you the proportion of true positives to the amount of total positives that the model predicts. It answers the question **"Out of all the positive predictions we made, how many were true?"**

Recall focuses on how good the model is at finding all the positives. Recall is also called true positive rate and answers the question "**Out of all the data points that should be predicted as true, how many did we correctly predict as true?**"

F1 Score is a measure that combines recall and precision. As we have seen there is a trade-off between precision and recall, F1 can therefore be used to measure how effectively our models make that trade-off.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$



[What is Accuracy, Precision, Recall and F1 Score?]

Understanding Classification: Results

Acquire → Learning

Memory
Adaptability

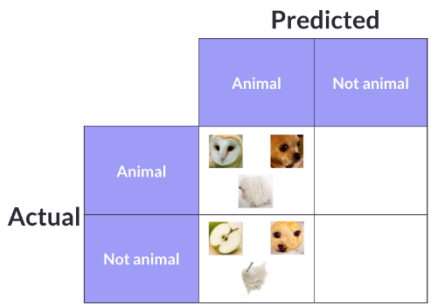
Tacit | Explicit | Implicit

Apply

Think, plan & reason

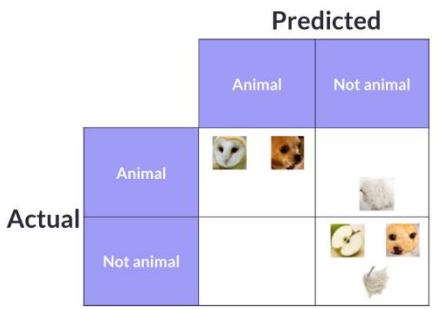
Problem → Solution

Classifies all images as animal



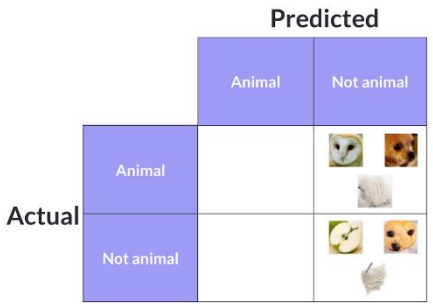
True Positives	3	
True Negatives	0	
False Positives	3	
False Negatives	0	
Accuracy	50%	$\frac{3+0}{3+0+3+0}$
Precision	50%	$\frac{3}{3+3}$
Recall	100%	$\frac{3}{3+0}$
F1 score	67%	$2 \cdot \frac{0.5 \cdot 1}{0.5+1}$

Overpredicts Images as not animal



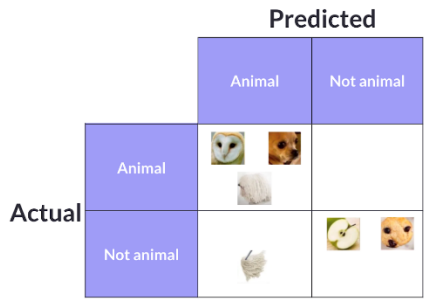
True Positives	2	
True Negatives	3	
False Positives	0	
False Negatives	1	
Accuracy	83%	$\frac{2+3}{2+3+1+0}$
Precision	100%	$\frac{2}{2+0}$
Recall	67%	$\frac{2}{2+1}$
F1 score	80%	$2 \cdot \frac{1 \cdot 0.67}{1+0.67}$

Classifies all images as not animal



True Positives	0	
True Negatives	3	
False Positives	0	
False Negatives	3	
Accuracy	50%	$\frac{0+3}{0+3+3+0}$
Precision	*100%	$\frac{0}{0+0}$
Recall	0%	$\frac{0}{0+3}$
F1 score	0%	$2 \cdot \frac{1 \cdot 0}{1+0}$

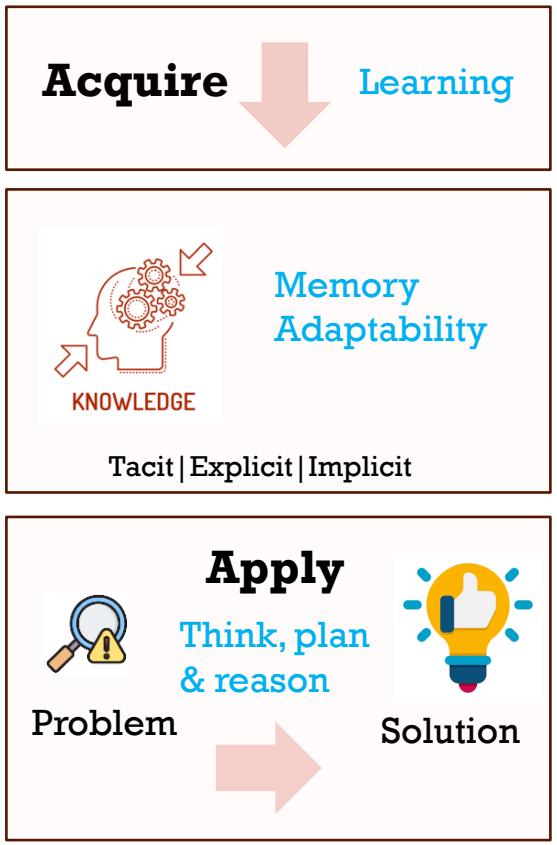
Overpredict images as animal



True Positives	2	
True Negatives	3	
False Positives	0	
False Negatives	1	
Accuracy	83%	$\frac{3+2}{3+2+0+1}$
Precision	75%	$\frac{3}{3+1}$
Recall	100%	$\frac{3}{3+0}$
F1 score	86%	$2 \cdot \frac{0.75 \cdot 1}{0.75+1}$

[What is Accuracy, Precision, Recall and F1 Score?]

Understanding Classification: Results



Feature	Precision (Quality)	Recall (Coverage)
Focus	Reducing **False Positives** (False Alarms)	Reducing **False Negatives** (Missed Cases)
Question	"Of all patients predicted to have a disease, how many actually did?"	"Of all patients who actually have the disease, how many did we find?"
Formula	True Positives / (True Positives + False Positives)	True Positives / (True Positives + False Negatives)
Clinical Goal	Avoid unnecessary, costly, or harmful treatments.	Avoid missing a serious, treatable condition.
Example	An AI flagging 100 people for cancer, where 90 actually have it (90% precision).	An AI finding 90 out of 100 people who actually have cancer (90% recall).

What is Accuracy, Precision, Recall and F1 Score?

Understanding Classification: Examples

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

CLASSIFICATION

Predict a class label
(Yes/No, Type A/Type B,
Category 1/Category 2, etc.)

1. HEALTHCARE

Problem: Disease prediction
Goal: Predict whether a patient has a disease.

Input Features	Output Class
• Age, Gender • Symptoms • Lab test results • Medical history	Disease / No Disease

2. FINANCE / BANKING

Problem: Loan approval
Goal: Predict whether a loan application will be approved.

Input Features	Output Class
• Income • Credit Score • Employment Type • Existing Loans	Approved / Not Approved

3. E-COMMERCE / RETAIL

Problem: Customer purchase prediction
Goal: Predict whether a customer will purchase a product.

Input Features	Output Class
• Browsing history • Past purchases • Time on site • Demographics	Will Buy / Will Not Buy

4. MARKETING

Problem: Customer response prediction
Goal: Predict whether a customer will respond to a campaign.

Input Features	Output Class
• Age, Location • Past campaign response • Email clicks • Purchase behavior	Responded / Not Responded

5. MANUFACTURING

Problem: Product quality inspection
Goal: Classify product as defective or non-defective.

Input Features	Output Class
• Sensor readings • Temperature • Pressure • Machine parameters	Defective / Non-Defective

6. CYBERSECURITY

Problem: Threat detection
Goal: Classify network activity as normal or suspicious.

Input Features	Output Class
• IP address • Traffic size • Protocol type • Connection duration	Normal / Suspicious

7. EDUCATION

Problem: Student performance prediction
Goal: Predict whether a student will pass or fail.

Input Features	Output Class
• Attendance • Assignment score • Test score • Study hours	Pass / Fail

KEY TAKEAWAY: In every domain, we use historical data (input features with known labels) to train a model. The model learns patterns and predicts the class label for new, unseen data.

Understanding Clustering

Acquire Learning

Memory
Adaptability

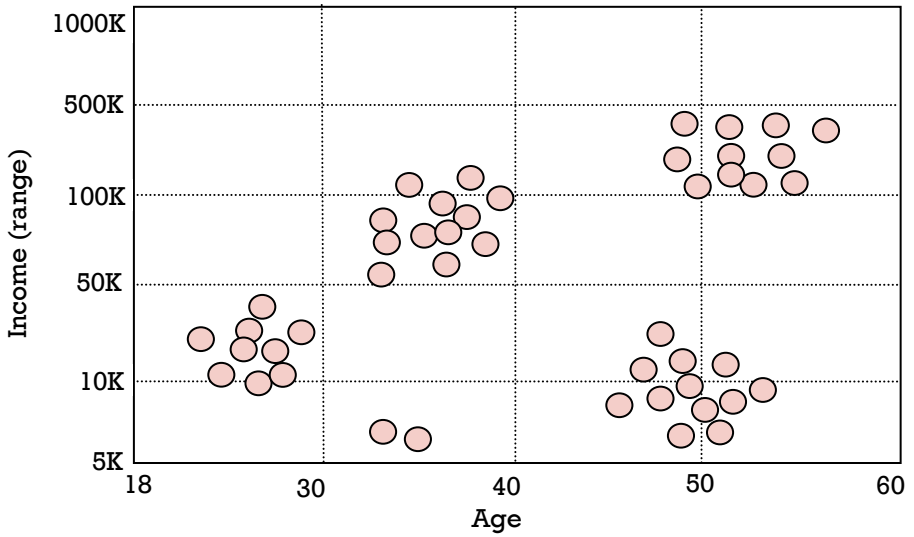
Tacit | Explicit | Implicit

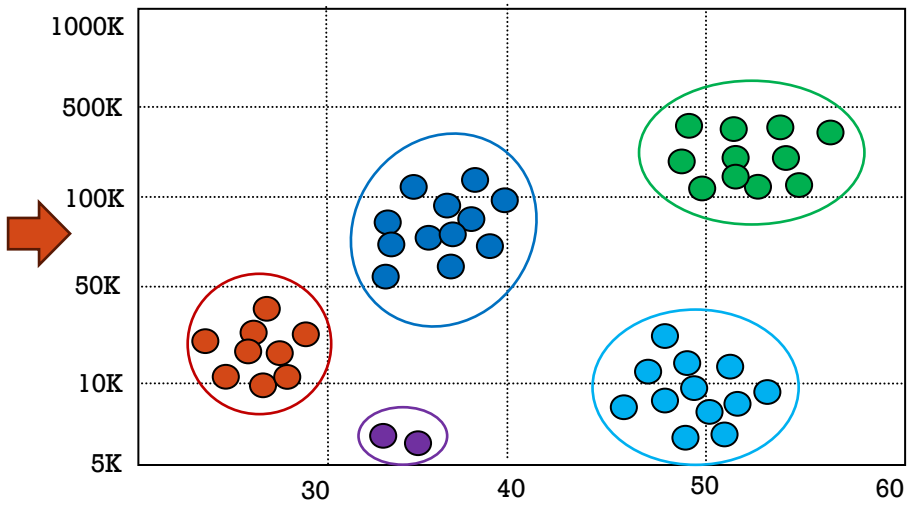
Apply

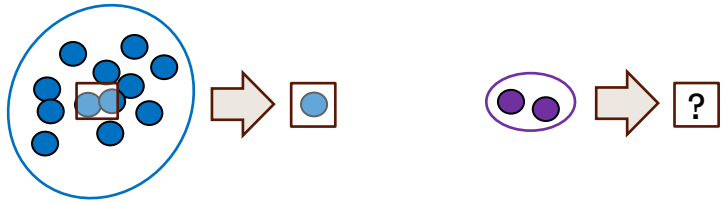
Think, plan & reason

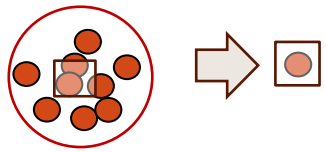
Problem

Solution









Understanding Clustering

Group customers based on their income and spending behavior.

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Customer Id	Annual Income	Spending Score
1	25000	20
2	28000	25
3	30000	22
4	45000	45
5	48000	50
6	50000	48
7	75000	75
8	78000	80
9	80000	78
10	90000	30
11	92000	28
12	95000	32

Feature	Meaning
AnnualIncome	Customer's annual income
SpendingScore	Score from 1 to 100 showing how much the customer spends

Understanding Clustering

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution

```python
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

data = [
    [1, 25000, 20],
    [2, 28000, 25],
    [3, 30000, 22],
    [4, 45000, 45],
    [5, 48000, 50],
    [6, 50000, 48],
    [7, 75000, 75],
    [8, 78000, 80],
    [9, 80000, 78],
    [10, 90000, 30],
    [11, 92000, 28],
    [12, 95000, 32]
]

df = pd.DataFrame(data, columns=[
    "CustomerId",
    "AnnualIncome",
    "SpendingScore"
])
```

```python
X = df[["AnnualIncome", "SpendingScore"]]

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

model = KMeans(n_clusters=4, random_state=7, n_init=10)
df["Cluster"] = model.fit_predict(X_scaled)

print(df)

plt.figure(figsize=(8, 5))

plt.scatter(
    df["AnnualIncome"],
    df["SpendingScore"],
    c=df["Cluster"]
)

plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.title("Customer Segmentation using K-Means Clustering")

plt.show()
```

K-Means is a clustering method that groups similar data points together. We first decide how many groups we want, called K. The algorithm then finds cluster centers and assigns each record to the nearest center. In customer segmentation, K-Means can automatically discover groups such as low-income low-spending customers, high-income high-spending customers, and high-income low-spending customers.

Understanding Clustering

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

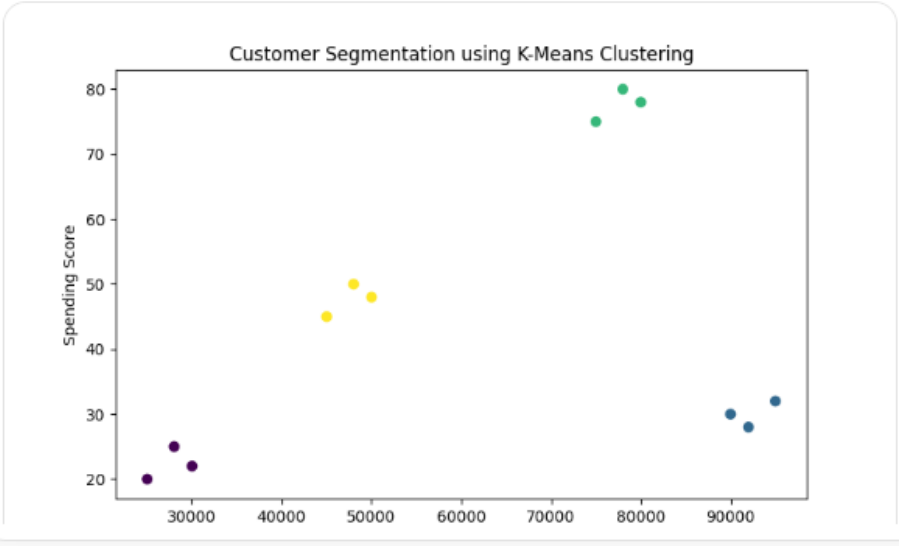
Apply

Think, plan & reason

Problem → Solution

Console

	CustomerId	AnnualIncome	SpendingScore	Cluster
0	1	25000	20	0
1	2	28000	25	0
2	3	30000	22	0
3	4	45000	45	3
4	5	48000	50	3
5	6	50000	48	3
6	7	75000	75	2
7	8	78000	80	2
8	9	80000	78	2
9	10	90000	30	1
10	11	92000	28	1
11	12	95000	32	1



Cluster	Business Interpretation
2	Low income, low spending customers
0	Medium income, medium spending customers
1	High income, high spending customers
3	High income, low spending customers

Cluster	Possible Action
Low income, low spending	Basic offers, discount schemes
Medium income, medium spending	Cross-selling and loyalty offers
High income, high spending	Premium products, priority service
High income, low spending	Special campaigns to increase engagement

Understanding Clustering: Examples

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

CLUSTERING

Discover natural groups / segments (Cluster 1, Cluster 2, Cluster 3, etc.) from unlabeled data.

1. HEALTHCARE

Problem: Patient segmentation
Goal: Group patients with similar health profiles.

Input Features:	Output Segments:
• Age • Symptoms • Lab values • Medical history	Chronic-risk / Moderate-risk / Low-risk

2. FINANCE / BANKING

Problem: Customer segmentation
Goal: Group customers with similar financial behavior.

Input Features:	Output Segments:
• Income • Spending pattern • Savings balance • Credit usage	High-value / Regular / At-risk customers

3. E-COMMERCE / RETAIL

Problem: Shopper segmentation
Goal: Group customers by buying behavior.

Input Features:	Output Segments:
• Purchase frequency • Basket value • Product categories • Browsing behavior	Frequent buyers / Bargain seekers / Occasional shoppers

4. MARKETING

Problem: Audience segmentation
Goal: Group users for targeted campaigns.

Input Features:	Output Segments:
• Demographics • Past responses • Click behavior • Channel preference	Highly engaged / Selectively engaged / Low engagement

5. MANUFACTURING

Problem: Machine or defect pattern grouping
Goal: Group products or machines with similar operating patterns.

Input Features:	Output Segments:
• Temperature • Vibration • Pressure • Cycle time	Normal operation / Warning pattern / Abnormal pattern

6. CYBERSECURITY

Problem: Network activity grouping
Goal: Group similar traffic patterns to detect unusual behavior.

Input Features:	Output Segments:
• IP activity • Protocol usage • Session duration • Traffic volume	Normal traffic / Suspicious cluster / High-risk cluster

7. EDUCATION

Problem: Student segmentation
Goal: Group students with similar learning behavior.

Input Features:	Output Segments:
• Attendance • Assignment scores • Test scores • Study time	High performers / Average learners / At-risk students

KEY TAKEAWAY: In clustering, there are no predefined class labels. The algorithm discovers similar groups in the data, and these groups can then be interpreted for decision-making thin.

Understanding Regression

Acquire Learning

Memory
Adaptability

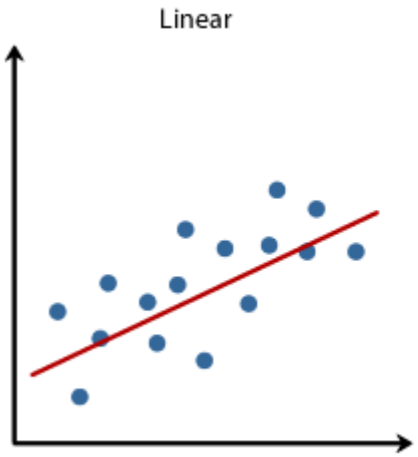
Tacit | Explicit | Implicit

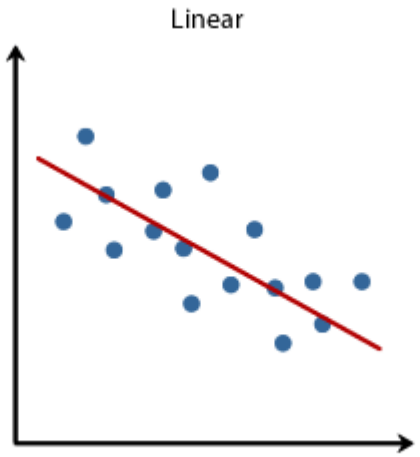
Apply

Think, plan
& reason

Problem

Solution





Understanding Regression: Examples

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

REGRESSION & FORECASTING

Predict numeric values such as sales, demand, cost, score, time, temperature, or future trends.

1. HEALTHCARE

Problem: Length-of-stay prediction / patient inflow forecast

Goal: Predict hospital stay duration or forecast patient arrivals.

Input Features:
- Age
- Symptoms severity
- Test results
- Historical admissions

Output:
- Days of stay / next-week patient count

2. FINANCE / BANKING

Problem: Revenue forecasting / loan loss prediction

Goal: Predict future revenue or expected loan loss.

Input Features:
- Income
- Repayment history
- Interest rate
- Monthly transactions

Output:
- Expected loss / next-month revenue

3. E-COMMERCE / RETAIL

Problem: Sales forecasting

Goal: Predict future product or store sales.

Input Features:
- Past sales
- Promotions
- Season
- Price

Output:
- Next-month sales

4. MARKETING

Problem: Customer lifetime value prediction

Goal: Predict numeric customer value or campaign conversions.

Input Features:
- Past purchases
- Frequency
- Channel usage
- Campaign spend

Output:
- Lifetime value / expected conversions

5. MANUFACTURING

Problem: Demand or maintenance-cost forecasting

Goal: Predict production demand or machine maintenance cost.

Input Features:
- Machine usage
- Temperature
- Vibration
- Historical demand

Output:
- Maintenance cost / production demand

6. ENERGY / UTILITIES

Problem: Electricity load forecasting

Goal: Predict power demand over time.

Input Features:
- Past load
- Temperature
- Day of week
- Holiday indicator

Output:
- Next-day load

7. EDUCATION

Problem: Score prediction / enrollment forecasting

Goal: Predict student marks or future enrollments.

Input Features:
- Attendance
- Study hours
- Assignment scores
- Historical admissions

Output:
- Exam score / next-year enrollment

KEY TAKEAWAY: Regression is used when the output is a number. Forecasting is a special predictive case focused on future numeric values using historical time-based data.

Understanding Association Rule Mining

Acquire → Learning

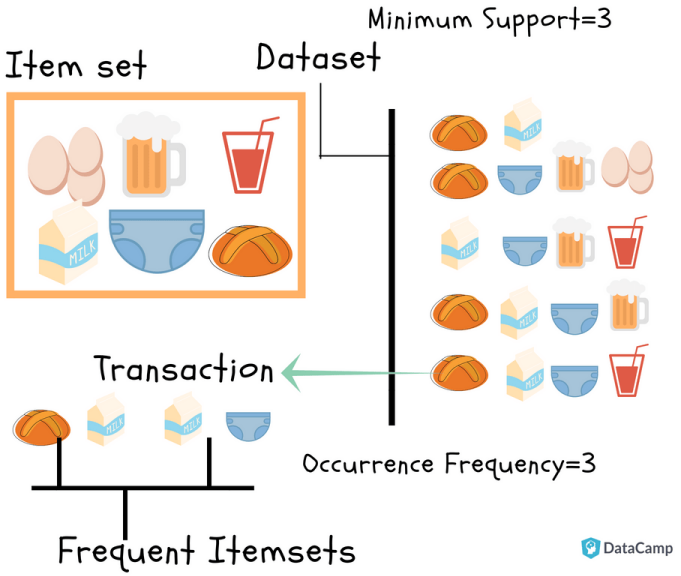
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



- ✓ Connecting entities and finding relationships
- ✓ Discovery
- ✓ More sales
- ✓ Inventory management
- ✓ Recommendation
- ✓ Cross-sell, up-sell

Understanding Filtering Techniques & P&R (personalization and recommendation)

Acquire → Learning

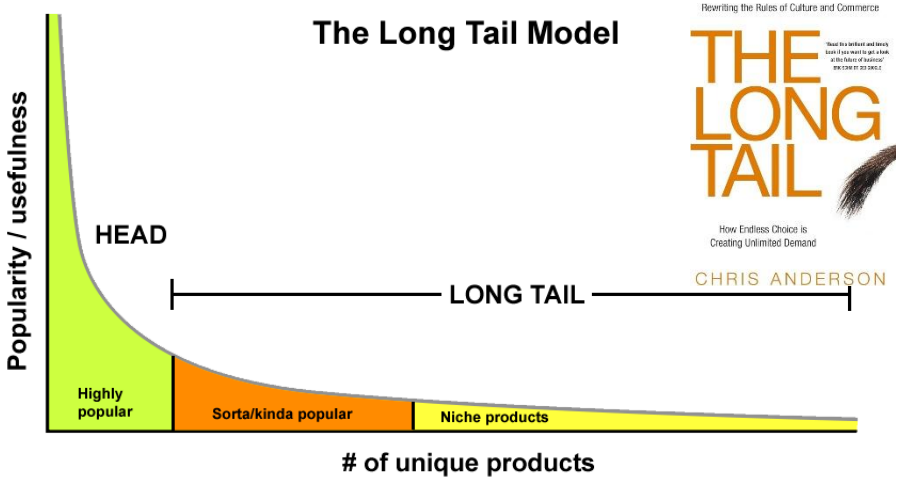
Memory
Adaptability

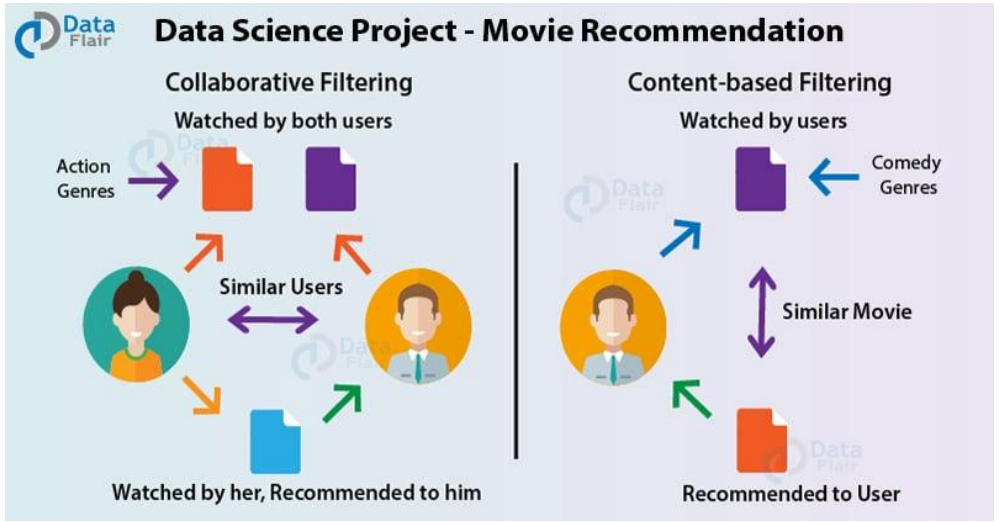
Tacit | Explicit | Implicit

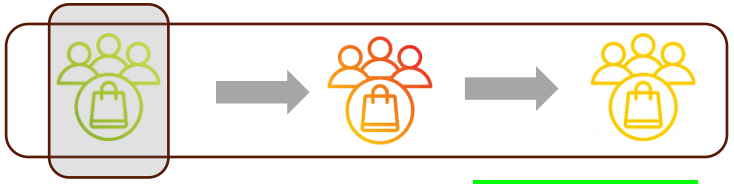
Apply

Think, plan & reason

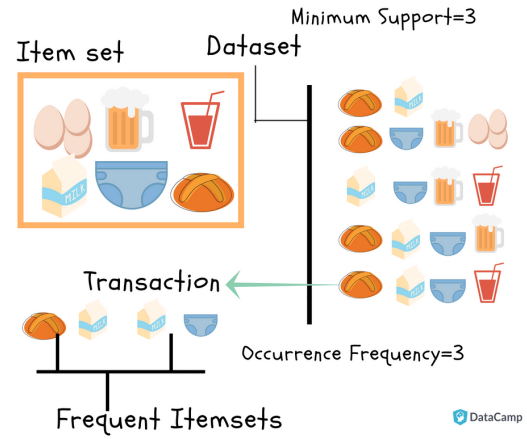
Problem → Solution







Move customers from buying highly popular to buying niche products to monetize.



- ✓ Discovery
- ✓ Reduce search time
- ✓ Increase customer loyalty
- ✓ More sales
- ✓ Content monetization
- ✓ Co-creation
- ✓ Inventory management
- ✓ Product placement
- ✓ Bundled offers
- ✓ New product
- ✓ Personalization

[Machine Learning Project - Data Science Movie Recommendation System Project in R - DataFlair]

Understanding Filtering Techniques

Collaborative filtering is a widely used technique in recommendation systems that predicts a user's interests by analyzing preferences and behaviors from similar users or items.

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Transactions		
User	Item	Rating
U1	I1	7
U1	I2	6
U1	I3	7
U1	I4	4
U1	I5	5
U1	I6	4
U2	I1	6
U2	I2	7
U2	I4	4
U2	I5	3
U3	I2	3
U3	I3	3
U3	I4	1
U3	I5	1
U4	I1	1
U4	I2	2
U4	I3	2
U4	I4	3
U4	I5	3
U4	I6	4
U5	I1	1
U5	I3	1
U5	I4	2
U5	I5	3
U5	I6	3

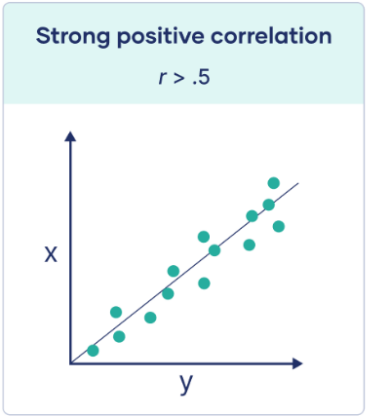
User↓ Item →	I1	I2	I3	I4	I5	I6	Mean	Pearson C. (r)
U1	7	6	7	4	5	4	5.5	0.894
U2	6	7		4	3	4	4.8	0.971
U3	?	3	3	1	1	?	2	1.000
U4	1	2	2	3	3	4	2.5	-1.000
U5	1		1	2	3	3	2	-0.866

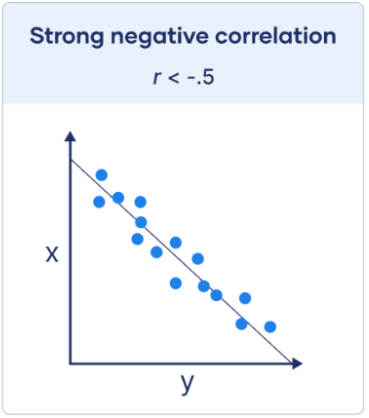
	By removing bias							
User↓ Item →	I1	I2	I3	I4	I5	I6		
U1	1.5	0.5	1.5	-1.5	-0.5	-1.5		0.894
U2	1.2	2.2		-0.8	-1.8	-0.8		0.971
U3	?	1	1	-1	-1	?		1.000
U4	-1.5	-0.5	-0.5	0.5	0.5	1.5		-1.000
U5	-1		-1	0	1	1		-0.866

U3 will rate I1	U3 will rate I6
6.48	4

By removing bias

U3 will rate I1	U3 will rate I6
3.34	0.86





Pearson's Correlation Coefficient: measures the strength and direction of the linear relationship between two continuous variables.

Understanding Filtering Techniques & P&R (personalization and recommendation)

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Problem

Think, plan & reason

Solution

Item-based filtering

	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

Understanding Filtering Techniques & P&R (personalization and recommendation)

Acquire → Learning

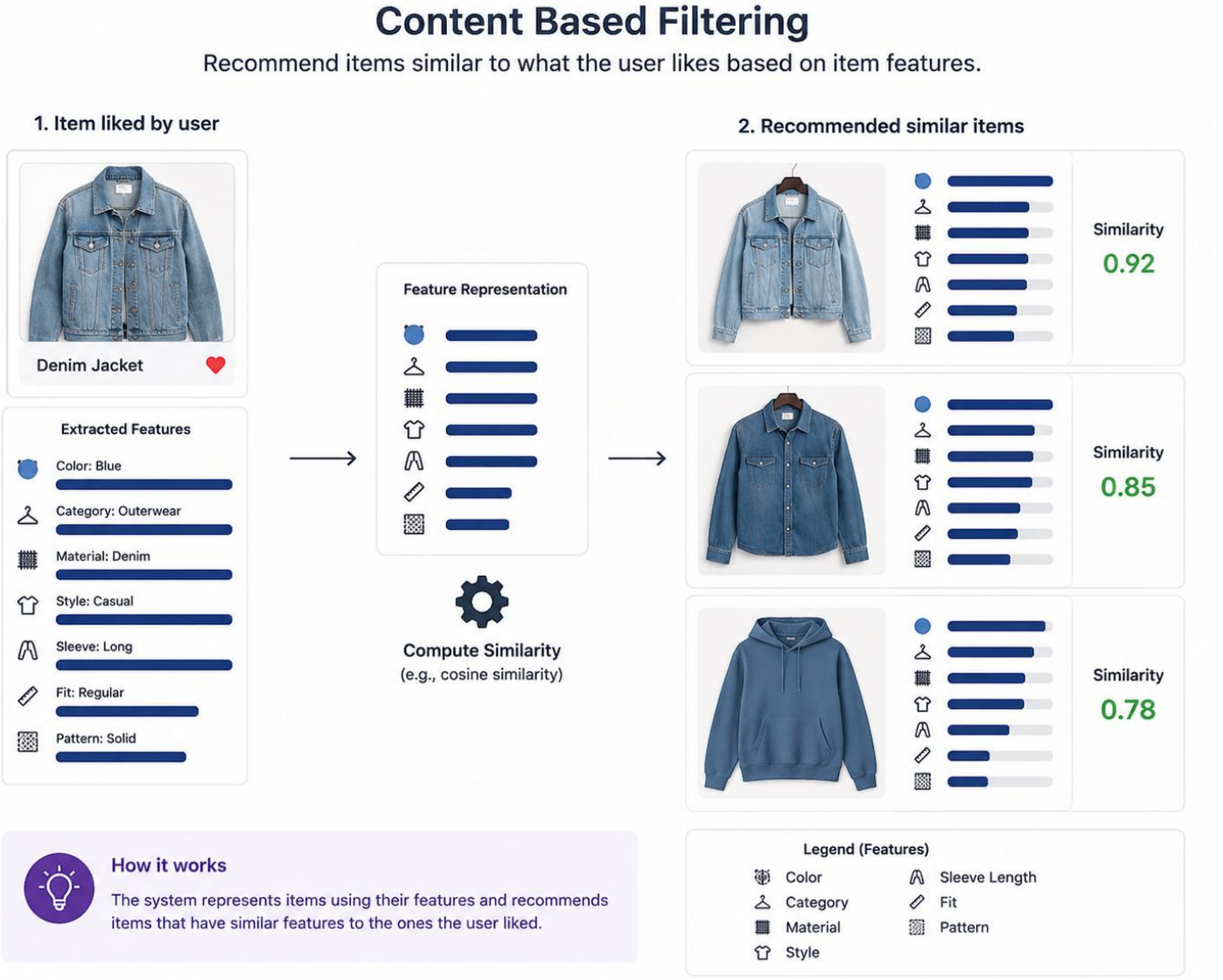
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Recommender Systems An introduction, Dietmar Jannach, TU Dortmund, Germany

How Netflix uses ML*

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

What are the generic entities and data ?

Netflix


Contents


Viewers


View, Rate, add to wish list, etc.


Devices


Contents screen


SMS | WhatsApp | Email

What can be recommended to the viewer based on what the user has watched, liked or currently displayed? What to be shown to new user?

How entities interact with each other?







Who watched (id, profile)? what content (content id)? when (time)? from where (location), how (duration, skips, intermittent, etc.) ? on what device (type, resolution etc.)? how content was found (search, recommendation?), by what channel in-app, email, sms, etc.

What can be done in the context of each entity?

How to monetize existing contents?

Create new contents (produce or acquire them) which are going to be viewed by large subscriber base? Efficiently (least cost)?

What contents are seen often, when and where? What is overall trend?

What data is required?

What is next the viewer is likely to watch?

What can be recommended to the viewer based on what the user has watched, liked or currently displayed?

What to be shown to new user?

How to customize content thumbnails to attract viewers?

How customer view time can be increased and customer loyalty?

What data is needed?


How can increase experience of viewers using different devices and band widths? Can there be plans based on content consumption (resolution, etc.)

Which is the best channel to reach viewers for recommendation?

What data is needed?

What AI/ML techs can be implement? and how?

What are the business benefits, What is ROI?

*Disney+ and Machine Learning in the Streaming Age

How Netflix uses ML: Excerpts

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Netflix relied heavily on ML to drive its growth and retain subscribers—with great success. (See Exhibit 3 for Netflix's growth trajectory from 2007 to 2016.) Most notably, it used ML to generate content recommendations, which accounted for an estimated 80 percent of the TV shows that people watched on the service.[16]

Netflix also assigned staff to watch shows and tag them, scene by scene, with a range of attributes, including details about the plot, the setting, and the cast. Then it combined these tags with the behavioral data and used ML to determine the importance of each data point in generating recommendations.

How much should it matter if a consumer watched something yesterday? Should that count twice as much or ten times as much compared [with] what they watched a whole year ago? How about a month ago? How about if they watched ten minutes of content and abandoned it or they binged through it in two nights? How do we weigh all that

What those things create for us is 'taste communities' around the world. It's about people who watch the same kind of things that you watch.

"Because Netflix is a data company, they know exactly how their viewers watch things," Fukunaga says. "So they can look at something you're writing and say, We know based on our data that if you do this, we will lose this many viewers. So it's a different kind of note-giving. It's not like, Let's discuss this and maybe I'm gonna win. The algorithm's argument is gonna win at the end of the day. So the question is do we want to make a creative decision at the risk of losing people."

Consequently, a movie or show's potential to be a blockbuster was less important than its power to connect with relatively small taste communities. That kind of capacity for precision matchmaking could, in general, create great value for both consumers and businesses, according to the authors

The image that was most likely to motivate that particular viewer to try the show. "In the end," a Netflix analyst wrote, "we saw one clear thing: Using better images to represent content significantly increased overall streaming hours and engagement from our members.

Extracts from: Disney+ and Machine Learning in the Streaming Age

Techniques | Technologies | Algos

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Each of these core technique, algo or tech are based on certain foundation, principle and concept, can address wide range of generic problems discussed before. Have capabilities and limitations and requirement of data and knowledge can vary greatly. There are several factors on which specific technique can be selected, modelled and implemented.

Code — Normal Programming Code

- Smart coders can include lot of intuitive and common sense for **operational intelligence** while writing the code in UI, workflows, processes etc.

Algos — Classical AI algorithms

- **Programs/Algorithms** to reach/search solutions e.g. min max algo pruning

KBS — Knowledge Based Systems

- Expert and Fuzzy Systems: model explicit **human** domain **knowledge** and solve problems using that expertise

ML — Data driven ML

- Data driven, **learn** relationships, patterns, associations etc. ANNs model human brain

GA — Genetic Algorithms

- Based on evolution principle: **evolve** solution to the problem, classified as evolutionary computing based on Charles Darwin's principle: survival of the fittest.

CBR — Case Based Reasoning

- Models human problem solving: remember, recall, adapt, reuse past **experiences,** lazy learning

MBR — Model Based Reasoning

- Expects things to work according to **modelled behaviour**. Monitors/detects deviation and helps in diagnosis

RL — Reinforcement Learning

- Self learning kind. Makes decisions to achieve the most optimal results. It mimics the **trial-and-error learning process** that humans use to achieve their goals.

Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

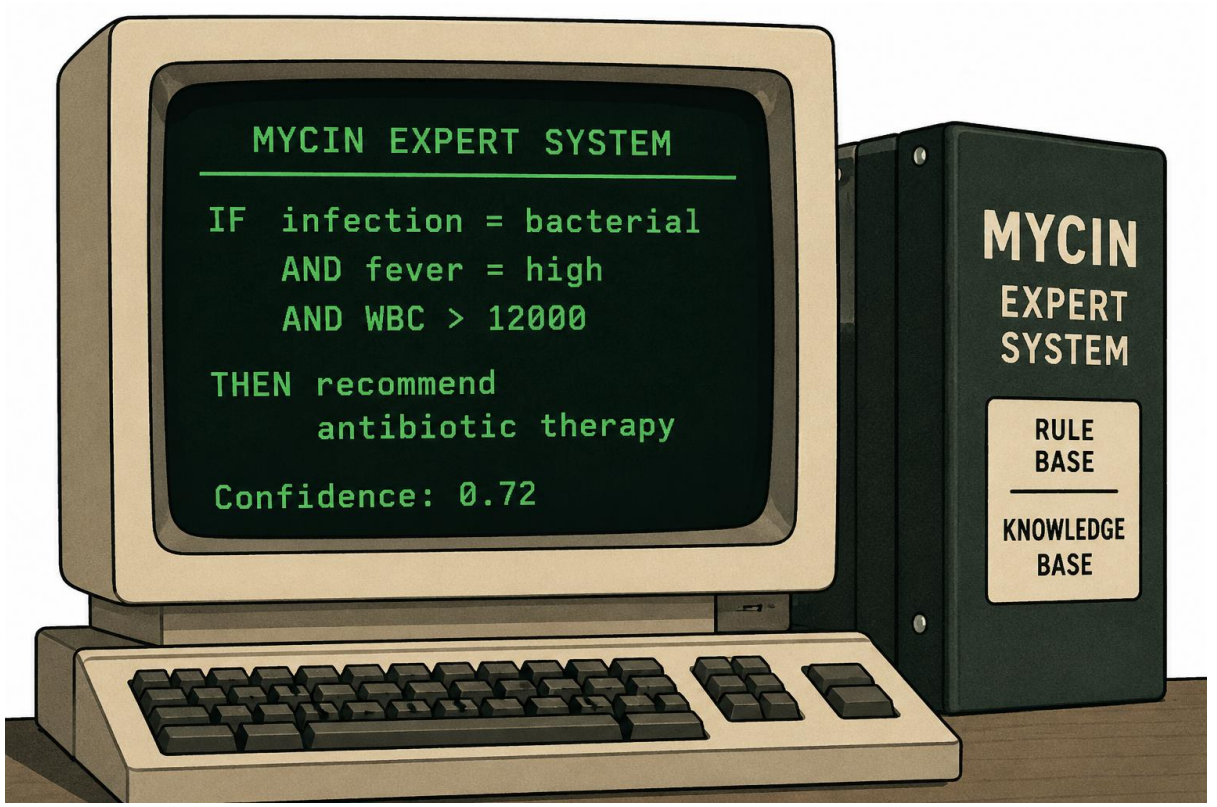
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

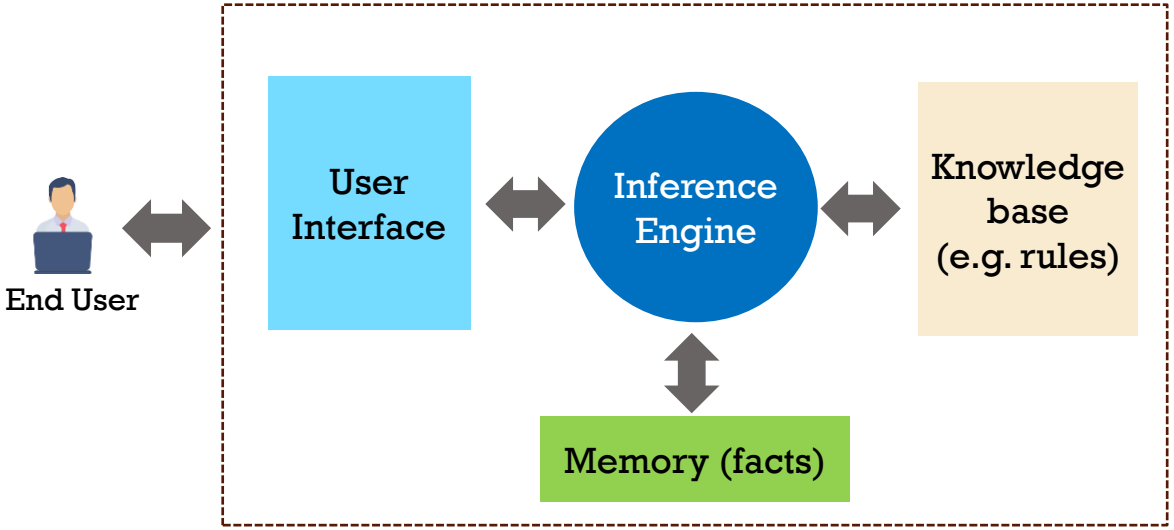
Think, plan & reason

Problem

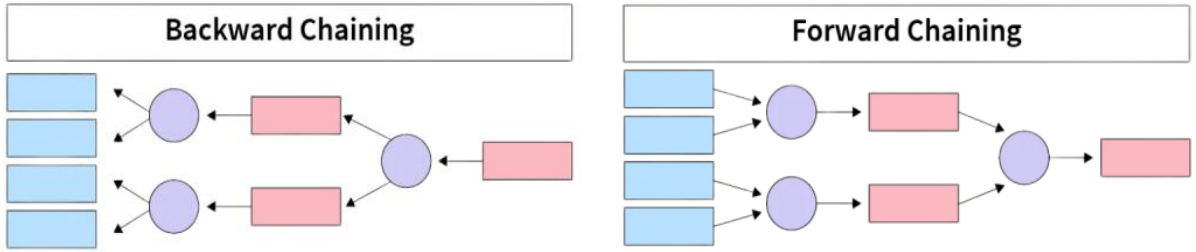
Solution

Mimics the intelligence of domain expert(s)

Basic Expert System Components



- ✓ Expert systems work like intelligent and smart agents by mimicking intelligence of human experts (no hallucinations and credible knowledge).
- ✓ Inference engine and knowledge base are separate making it easier to manage dynamic business logic.
- ✓ Ask only relevant inputs based on the context and heuristics
- ✓ Rules are easier to understand. Outcomes can be explained (explainability of reasoning), takes care of regulatory and compliance requirements.



Rule-based reasoning

Case Study: Countrywide Loan Underwriting Expert System (CLUES)

Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

— Example: Automated Medical Diagnosis —

You interact with the system through a conversation. It asks questions, records answers, reasons with medical rules, and provides explainable advice.

THE CORE RATIONALE

When expert knowledge can be captured as rules, a system can consistently ask the right questions, apply those rules to your answers, and provide explainable, reliable recommendations.

1 Patient Presents Symptoms

Patient provides symptoms and findings.

2 Capture Facts

Symptoms, age, vitals, lab findings are entered into the system.

- Symptoms
- Age
- Vitals
- Lab Findings

3 Consult Knowledge Base

The system searches the rule base / knowledge base.

Sample Rules (IF–THEN)

IF fever AND productive cough AND chest pain THEN consider pneumonia

IF sore throat AND mild fever THEN consider viral infection

IF bacterial infection suspected THEN recommend antibiotic therapy

Knowledge Base (Rule Base)

4 Inference Engine Applies Rules

Matches facts to rules and derives a likely diagnosis.

Matching Facts

Fever	✓
Productive Cough	✓
Chest Pain	✓
Age	35 yrs
WBC Count	12,800 /µL

Derived Conclusion

Likely Diagnosis: Pneumonia

5 Recommendation & Explanation

System provides diagnosis, suggested treatment, and explains why.

Diagnosis
Pneumonia

Suggested Treatment
- Antibiotics
- Rest & hydration
- Follow-up in 3–5 days

Why this conclusion?
Matched Rule #1 because fever + productive cough + chest pain are present.

6 Doctor Reviews & Decides

Doctor validates the suggestion, considers context, and finalizes treatment.

Final Decision
Treatment Plan Finalized

WHY EXPERT SYSTEMS?

- Captures expert knowledge
- Gives consistent decisions
- Provides explanations
- Useful when domain rules are known

EXPERT SYSTEM

User / Doctor Interface
Doctor enters patient information and receives advice and explanations.

Knowledge Base (Rules)
Stores expert knowledge as IF–THEN rules and medical facts.

Inference Engine
Applies rules to the facts using reasoning to derive conclusions.

Explanation Module
Generates explanations for the conclusions and recommendations.

LIMITATION

Best suited when knowledge can be articulated explicitly as rules; less flexible for novel or ambiguous situations.

Solve problems by codifying expert knowledge into rules and applying them consistently.

Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution

Expert systems in Q&A mode*

Search.Product Type

Welcome, tell us what you would like to do?

- (•) Would you like to keep your money with us?
- ( ) Are you in need of some money?

OK Back

Break Session

Search.Deposit Type

Ok, you are welcome to keep money with us. What are your plans?

- ( ) You want your money back at end of some period?
- ( ) You need some money (interest) to take care of your periodic expenses?
- (•) You want to put some money every month with us?

OK Back

Break Session

Search.Installment Amount

How much money you want deposit every month?

1000

OK Back

Break Session

Search.Deposit Period

Please specify exact number of: Months of your investment

12

OK Back

Break Session

Search.Variable Installment

Would the money (that you deposit) vary every month?

- ( ) Yes
- (•) No

OK Back

Break Session

Report

The product type selected	**Deposit**
The sub type of the product	**Recurring**
Total amount you have invested/will invest	**Rs.12000 (12*1000)**
Total period of your investment	**12 Months**
Total amount you will get at the end of period:	**Rs.12329**

As a total interest you will recieve an amount of **Rs.329** at the end of **12 Months**
View Matching/Suggested Products

Product/s matching your criteria

S.No.	BankProduct.Product_Code	BankProduct.Product_Name
1	030	PROGRESSIVE DEPOSITS

On Applications of Web-Based Intelligent Systems-Focus Banking Applications, Oorja, Jan-April 2005

Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

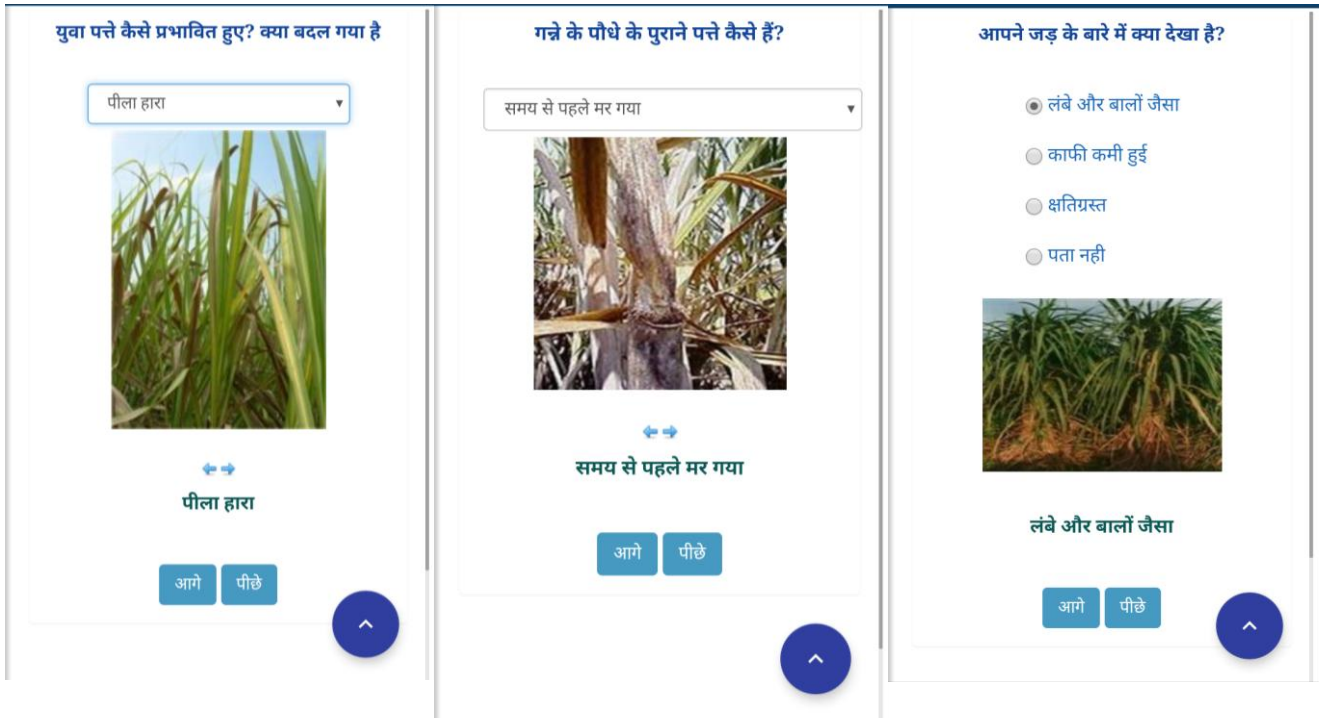
Apply

Think, plan & reason

Problem

Solution

Expert systems in Q&A mode*



Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution

Expert systems in Q&A mode*

Example: Knowledge-based systems: Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Rule 6: Phosphorus Deficiency — SugarcaneApp

IF

```
Young leaf IS greenish blue but narrow
AND Root IS decreased considerably
AND Plant IS Short
```

THEN

```
Deficiency Observed IS Phosphorus
```

Rule 7: Nitrogen Deficiency — SugarcaneApp

IF

```
Young leaf IS yellow green
AND Older leaf IS drying at premature stage
AND Girth IS reduced
AND Root IS Long and hairy
AND Plant IS Short
```

THEN

```
Deficiency Observed IS Nitrogen
```

Rule 8: Magnesium Deficieny — SugarcaneApp

IF

```
Young leaf IS red necrotic and rusty
AND Older leaf IS yellow at edges & green at center
AND Chlorotic IS at the tip and margins
AND Stalk IS affected with Internal browning
```

THEN

```
Deficiency Observed IS Magnesium
```

Rule 9: Boron Deficiency — SugarcaneApp

IF

```
Young leaf IS distorted leaf
AND Meristem IS died
AND Chlorotic IS on the leaf in variable amount
AND Tiller IS bunched with young plants
```

THEN

```
Deficiency Observed IS Boron
```

Rule 10: Calcium Deficiency — SugarcaneApp

IF

```
Young leaf IS ANY OF [distorted leaf,red necrotic and rusty]
AND Older leaf IS having a rusty appearance and may die
AND Chlorotic IS on whole leaf
```

THEN

```
Deficiency Observed IS Calcium
```

Rule 11: Copper Deficiency — SugarcaneApp

IF

```
Young leaf IS leaf of green splotches
AND Meristem IS alive
AND Plant IS Short
```

THEN

```
Deficiency Observed IS Copper
```

Rule 12: Sulphur Deficiency — SugarcaneApp

IF

```
Young leaf IS red necrotic and rusty
AND Chlorotic IS on whole leaf
```

THEN

```
Deficiency Observed IS Sulphur
```

Rule 13: Iron Deficiency — SugarcaneApp

IF

```
Young leaf IS yellow green
AND Root IS damaged
```

THEN

```
Deficiency Observed IS Iron
```

Example: Classification using Rule-based classifier

Acquire → Learning

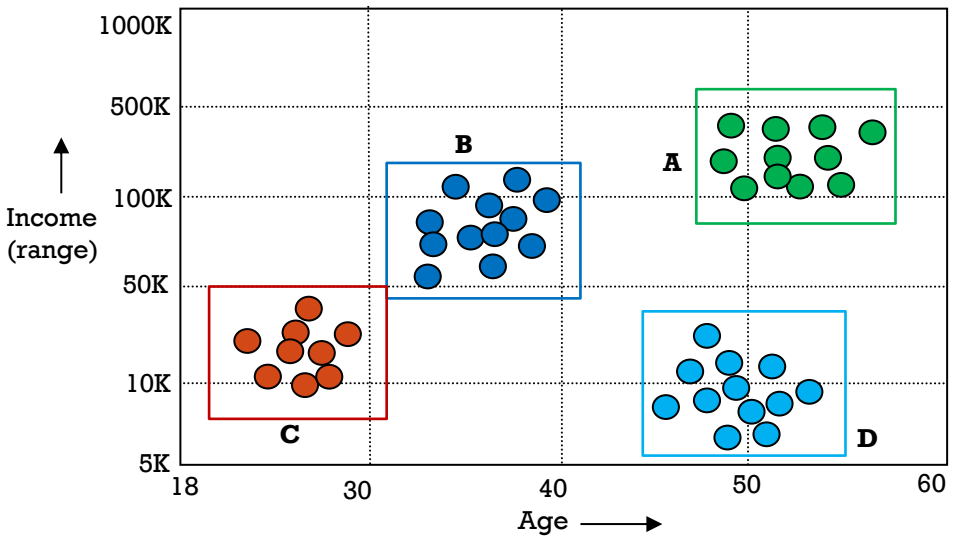
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Customers are classified into four groups based on features: A,B,C & D

Rule 1: Class A
If income between 75K to 550K
and age is between 47 to 58
then customer belongs to class A

Rule 2: Class B
If income between 48K to 250K
and age is between 31 to 42
then customer belongs to class B

Rule 3: Class C
If income between 7.5K to 50K
and age is between 19 to 31
then customer belongs to class C

Rule 4: Class D
If income between 5K to 48K
and age is between 45 to 55
then customer belongs to class D

Example: Classification using Rule-based classifier

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

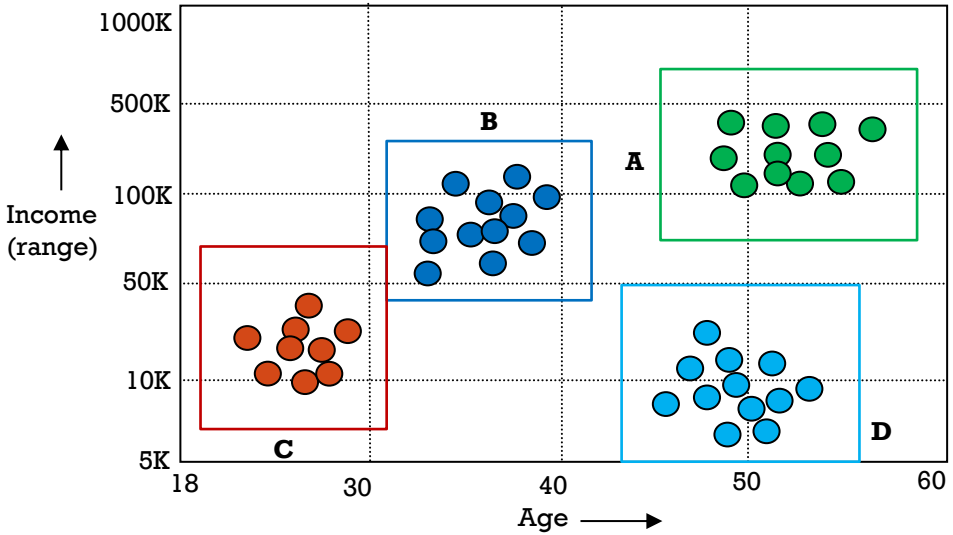
Apply

Problem

Think, plan & reason

Solution

Relaxed (generalized) Boundaries



Customers are classified into four groups based on features: A,B,C & D

Example: Classification using Case-based reasoning

Acquire → Learning

Memory
Adaptability

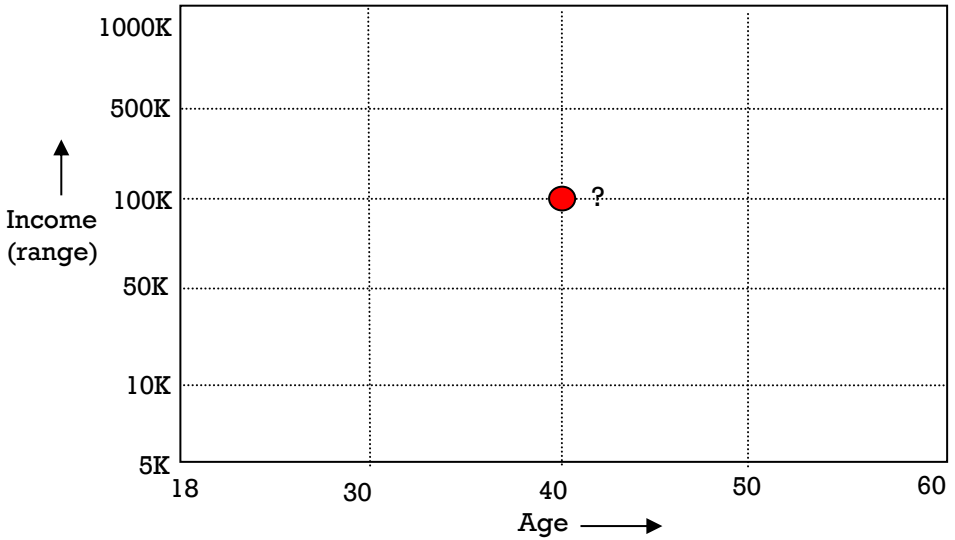
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution



Customer
Age:40
Income:100K
Class: ?

Which class this new customer would belong?

Rule 4: Class B
If income between 48K to 250K
and age is between 31 to 42
then customer belongs to class B

Class B

Example: Knowledge-based systems: Intelligent Agents using Expert System Tech

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution

Intelligent Agents: Theory and Practice

Michael Wooldridge
Department of Computing
Manchester Metropolitan University
Chester Street, Manchester M1 5GD
United Kingdom
M.Wooldridge@doc.mmu.ac.uk

Nicholas R. Jennings
Department of Electronic Engineering
Queen Mary & Westfield College
Mile End Road, London E1 4NS
United Kingdom
N.R.Jennings@qmw.ac.uk

Submitted to *Knowledge Engineering Review*, October 1994.
Revised January 1995.

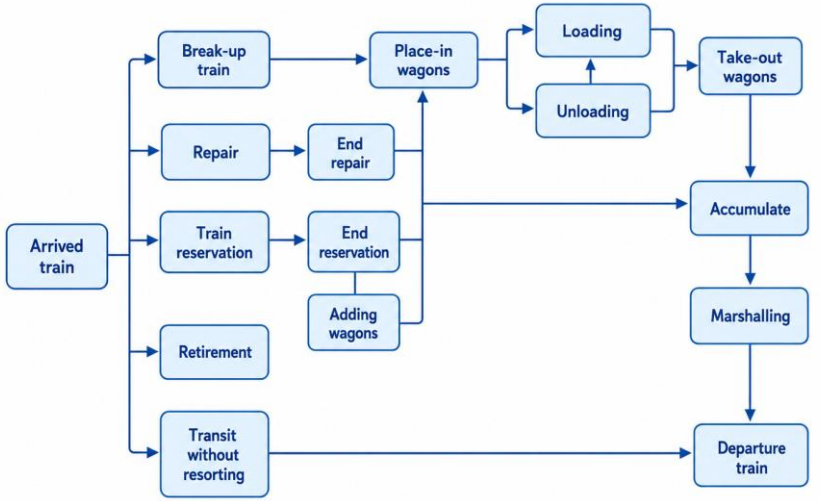
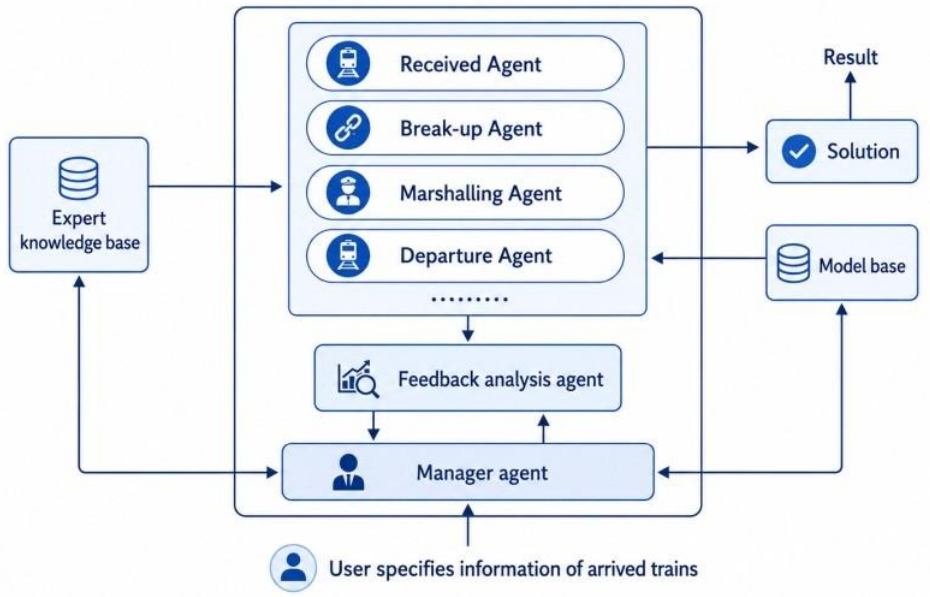


Fig. 2. Technique work operation process in marshalling station



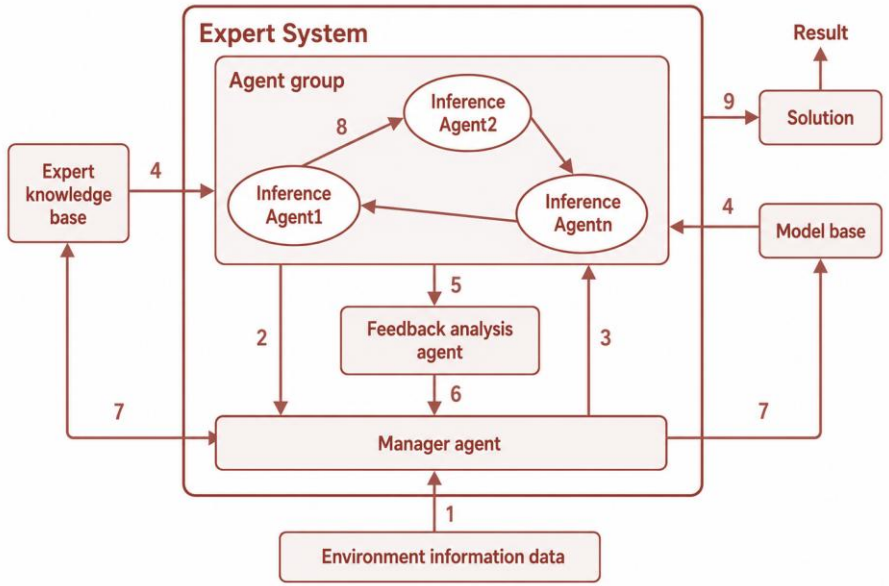


Fig. 1. IAES architecture

[Intelligent Agent-Based Expert System Architecture for Generating Work Plan Marshalling Station (2005)](#) *

[Intelligent Agents: Theory and Practice](#)

Note original diagrams are redrawn.

Example: Case-based reasoning

Acquire → Learning

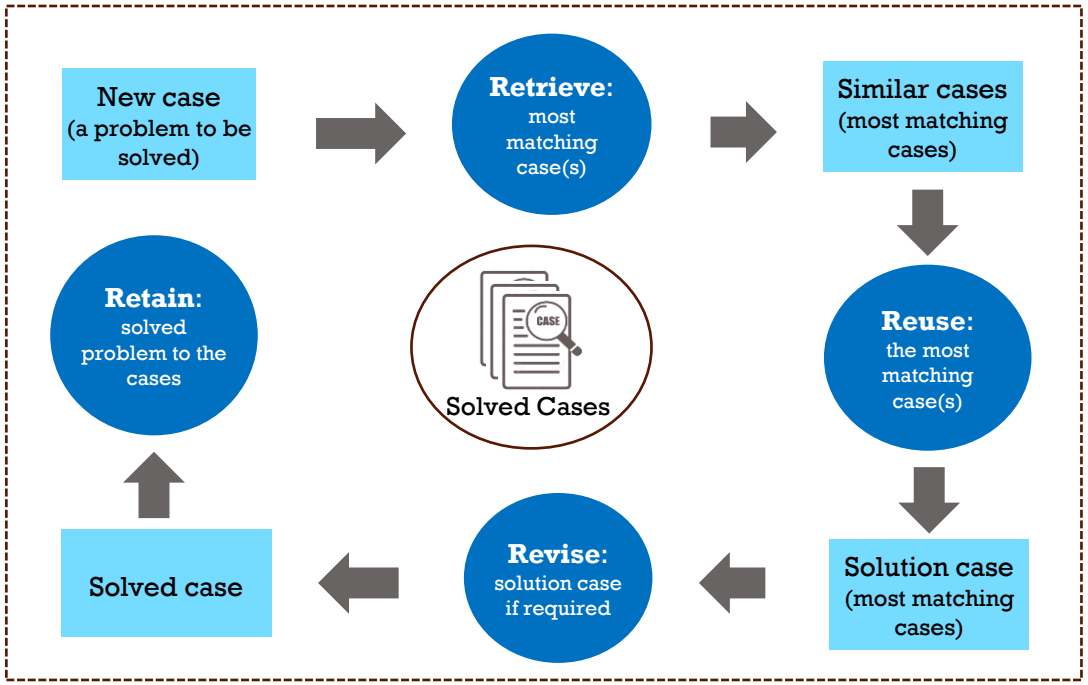
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Domain and meta-knowledge (ontologies, taxonomies etc.)

Domain Vocabulary	Feature weights & similarity measures	Similarities & computation logic	Reuse & revision logic

Generic structured case format

- ❑ Description/ID
- ❑ Specification
  Description, profile, symptoms, sequence of events...
- ❑ Solution
  Action, evaluation, decision, repair, diagnosis, remedy, prescription...
- ❑ Feedback*

* Is not part of standard case and is very important to understand effectiveness of prescribed solution.

- ✓ Supports dynamism (lazy learning) combining knowledge (semantics) and data driven intelligence.
- ✓ Each feature can be modelled based on requirements of domain.
- ✓ Models are easier to understand, customize and configure. Explainability.
- ✓ Can address large number of applications. Suitable for N=1 analysis and intelligence (e.g. hyper contextualization and personalization).

Credit status	Purchase behaviour	Churn analysis	Transaction monitoring	Product life	Monitoring/diagnostics
❑ Customer ID ❑ Customer profile: Age, gender, income, education, ... ❑ Repayment history: Good/Average...	❑ Customer ID ❑ Customer profile: Age, gender, income, education,... ❑ Items purchased: Item_1, Item_2, ...	❑ Customer ID ❑ Transaction details: Last recharge, recharge amount, friends status, change call usage,.. ❑ Churn status: Churned out	❑ Transaction ID ❑ Purchase details: Place, merchant, time and date, items purchased, amount, payment type,.. ❑ Payment status: OK/Delayed/Fraud	❑ Shelf product ID ❑ Product profile: name, category, price, store, location, discount offered, features,... ❑ Shelf life: Days in shelf	❑ Failure description ❑ Symptoms: S1(Q1-A1), S2(Q2-A2),... ❑ Fault/remedy: F1, R1 ...

Example: Case-based reasoning

Acquire → Learning

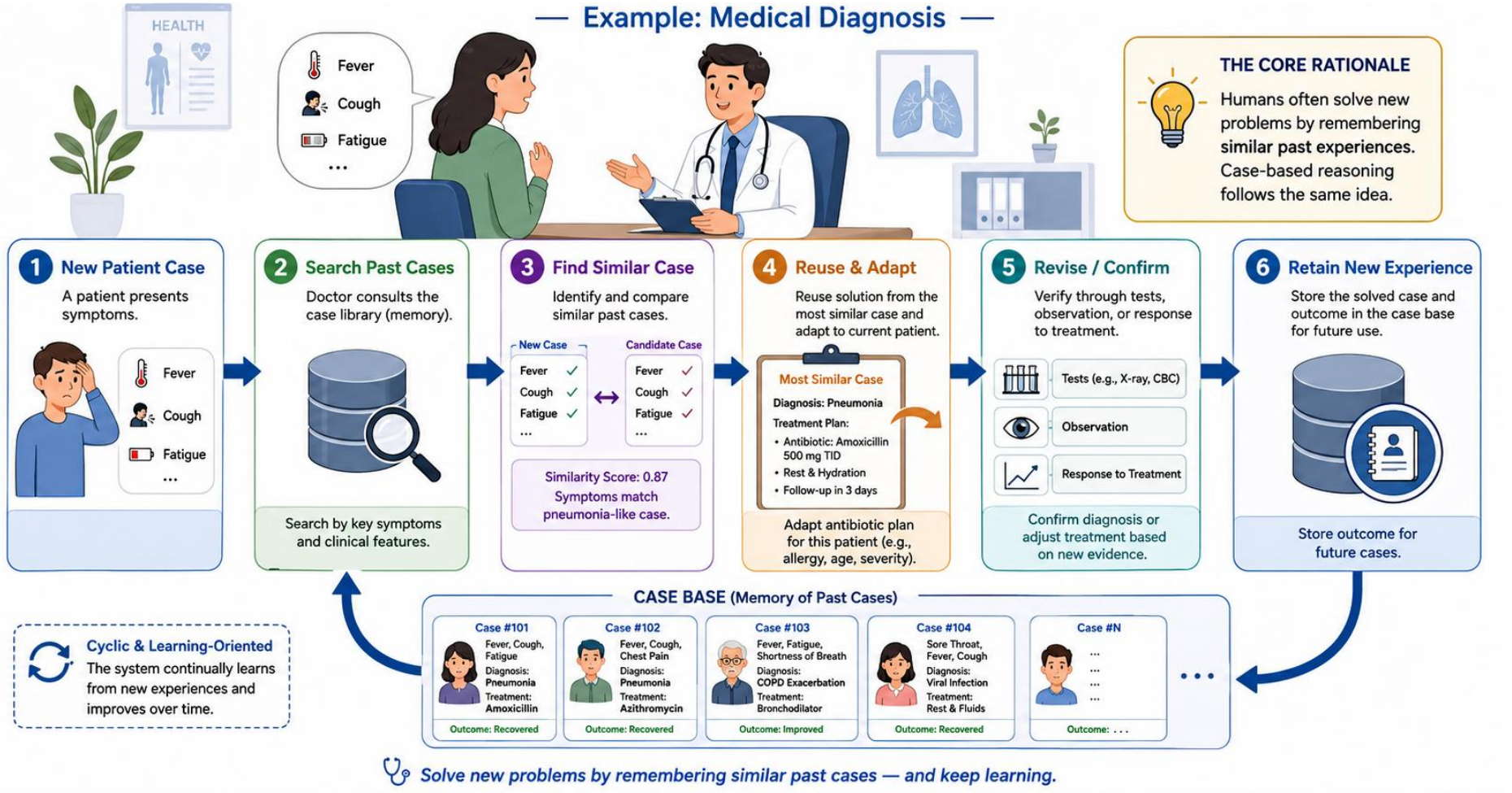
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Example: Case-based reasoning: Example

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Problem

Think, plan & reason

Solution

Case base

CASE 1	**Problem (Symptoms)** • *Problem*: Front light doesn't work • *Car*: VW Golf III, 1.6 l • *Year*: 1996 • *Battery voltage*: 13,6 V • *State of lights*: OK • *State of light switch*: OK **Solution** • Diagnosis: Front light fuse defect • Repair: Replace front light fuse
CASE 2	**Problem (Symptoms)** • *Problem*: Front light doesn't work • *Car*: Audi A4 • *Year*: 1997 • *Battery voltage*: 12,9 V • *State of lights*: surface damaged • *State of light switch*: OK **Solution** • Diagnosis: Bulb defect • Repair: Replace front light

Example: Case-based reasoning

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Problem case: case to be solved

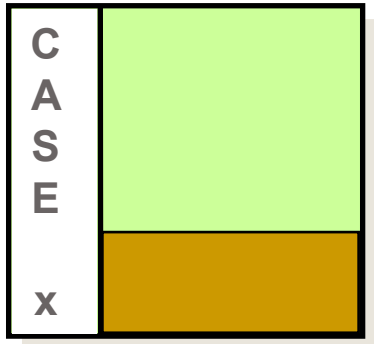
Feature

Value

Problem (Symptom):

- *Problem*: Break light doesn't work
- *Car*: Audi 80
- *Year*: 1989
- *Battery voltage*: 12.6 V
- *State of light*: OK

New Problem ↔ **Similar?** ↔



Example: Case-based reasoning

Acquire → Learning

Memory
Adaptability

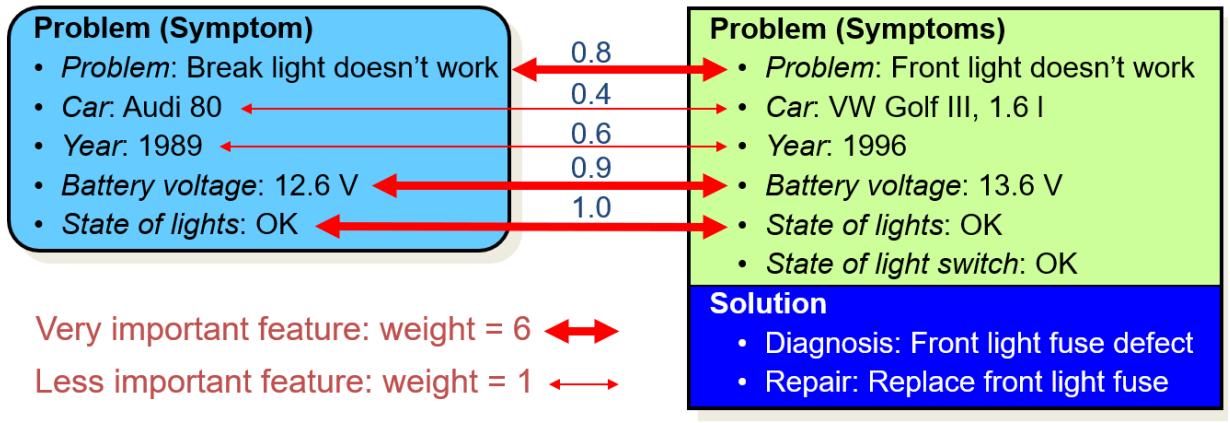
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

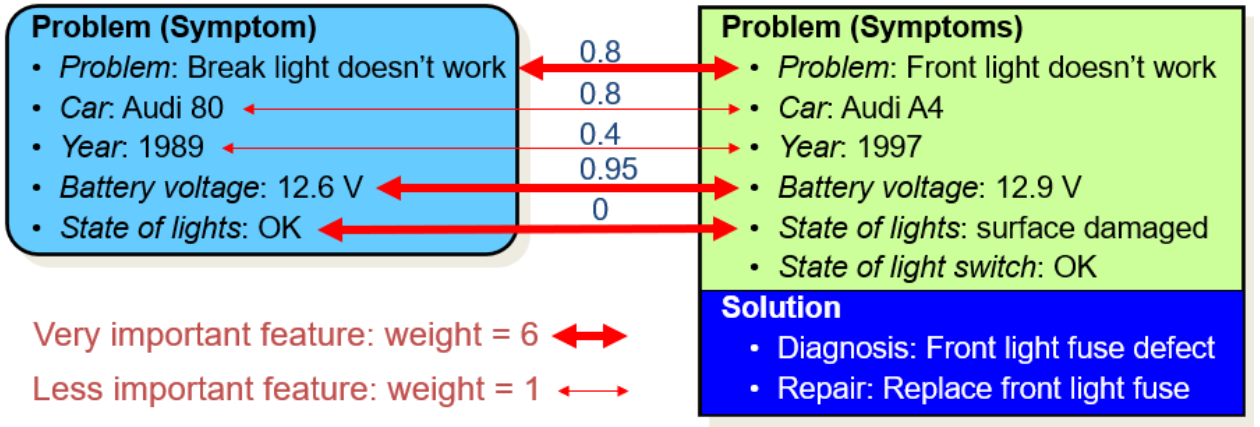
Finding nearest neighbour: retrieval



- Similarity computation by weighted average

similarity(new,case 1) = 1/20 * [6*0.8 + 1*0.4 + 1*0.6 + 6*0.9 + 6* 1.0] = **0.86**

Divided by 20 is to proportionate weights given to features (6+1+1+6+6)



- Similarity computation by weighted average

similarity(new,case 2) = 1/20 * [6*0.8 + 1*0.8 + 1*0.4 + 6*0.95 + 6*0] = **0.585**

Example: Case-based reasoning

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Adopting matching cases

CASE 1

Problem (Symptoms):
- *Problem*: **Front light** doesn't work
- ...

Solution:
- Diagnosis: **Front light** fuse defect
- Repair: Replace **front light** fuse

Problem (Symptom):
- *Problem*: Break light doesn't work
- *Car*: Audi 80
- *Year*: 1989
- *Battery voltage*: 12,6 V
- *State of light*: OK

Adapt Solution:

How do differences in the problem affect the solution?

New Solution:
- Diagnosis: **Break light** fuse defect
- Repair: Replace **break light** fuse

CASE 3

Problem (Symptoms):
- *Problem*: Break light doesn't work
- *Car*: Audi 80
- *Year*: 1989
- *Battery voltage*: 12.6 V
- *State of lights*: OK
- *State of light switch*: OK

Solution:
- Diagnosis: break light fuse defect
- Repair: replace break light fuse

Example: Classification using Case-based reasoning

Acquire → Learning

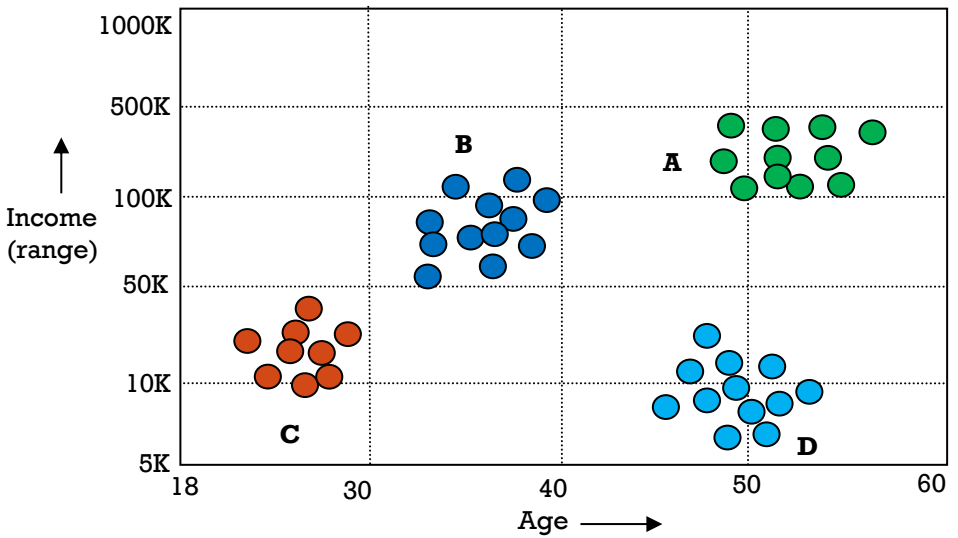
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Customers are classified into four groups based on features: A,B,C & D

Case structure

Problem description
Customer ID
Profile:
Age(Min:18,Max:60), Income of customer(Min 5K, Max:1000K)
Outcome:
Class to which the customer belongs (classified)

Problem Case:
Age:40
Income:100K
Class: ?

Which class this new customer would belong?

Example: Classification using Case-based reasoning

Acquire → Learning

Memory
Adaptability

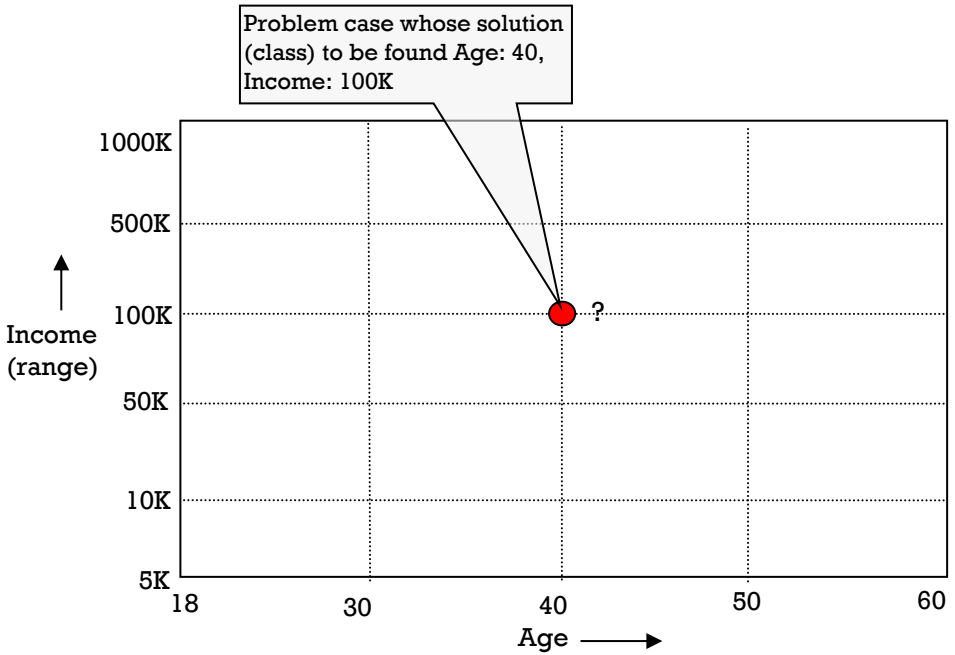
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution



Problem Case:
Age:40
Income:100K
Class: ?

Which class this new customer would belong?

Example: Classification using Case-based reasoning

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

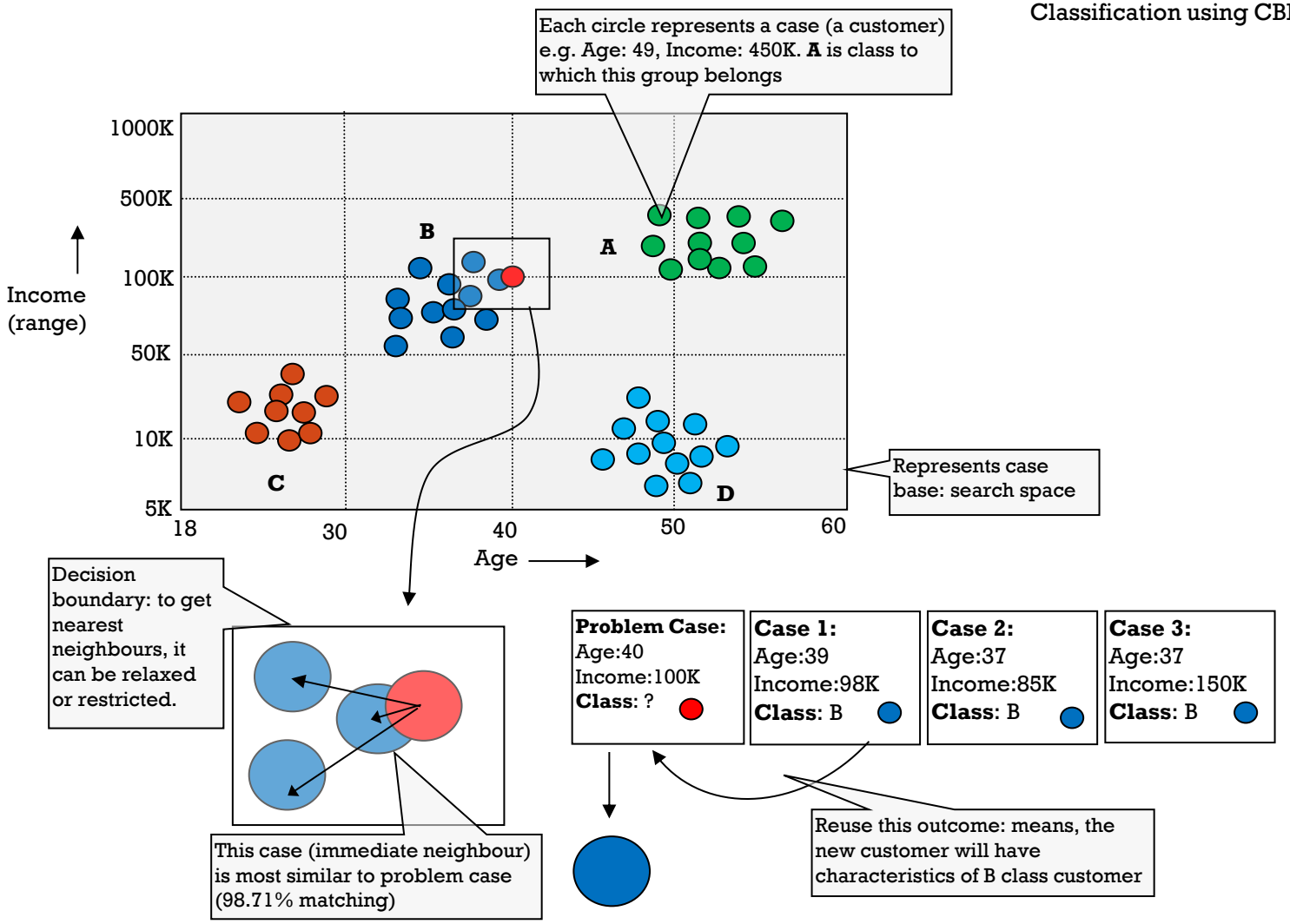
Apply

Think, plan & reason

Problem

Solution

Classification using CBR



Similarity computation for case 1

$Similarity(Age_p, Age_1) = 1 - |40-39|/(60-18) = 97.62\%$
$Similarity(Income_p, Income_1) = 1 - |100-98|/(1000-5) = 99.80\%$
$Similarity(Problem\ Case, Case1) = 0.5*97.62 + 0.5*99.80 = 98.71\%$

Example: Clustering using Case-based reasoning

Acquire → Learning

Memory
Adaptability

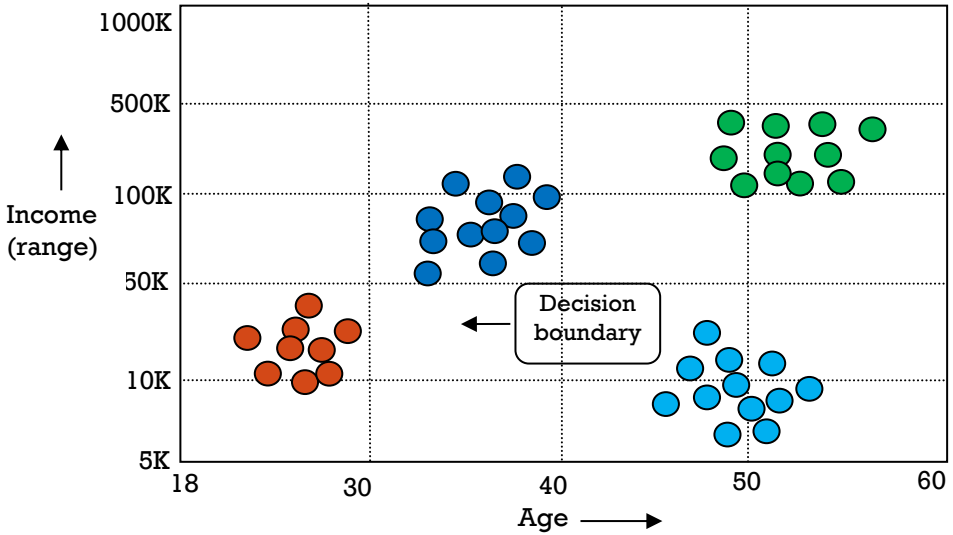
Tacit | Explicit | Implicit

Apply

Problem

Think, plan & reason

Solution



Example: Clustering using Case-based reasoning

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

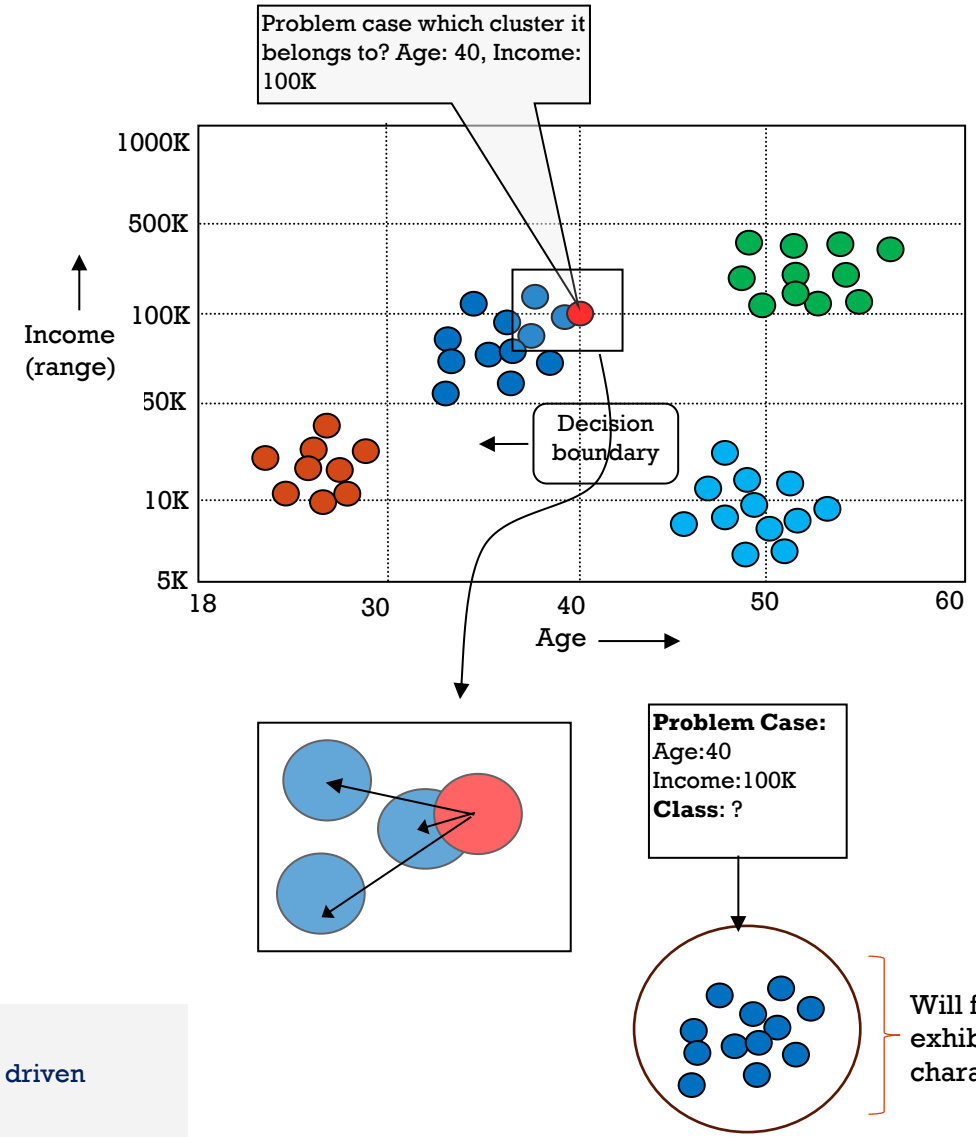
Think, plan & reason

Problem → Solution

Comparing ML driven classification/clustering v/s CBR driven

- ✓ Lazy learning
- ✓ Knowledge based + data driven learning
- ✓ Accuracy can be controlled
- ✓ Can focus on one entity at a time

Clustering using CBR



Problem description
Customer ID
Profile:
Age(Min:18,Max:60), Income of customer(Min 5K, Max:1000K)
Outcome:
Class to which the customer belongs (classified)

Example: Case-based reasoning

Acquire → Learning

Memory
Adaptability

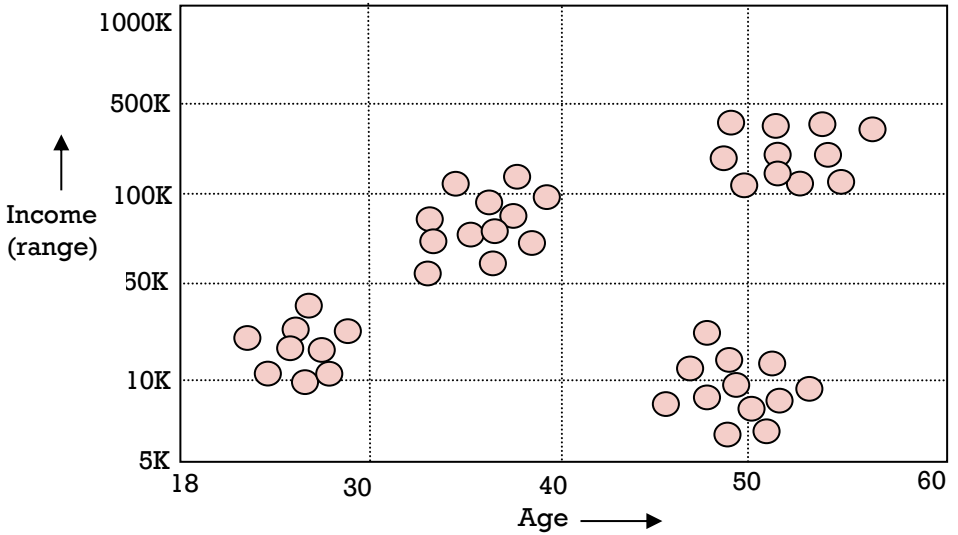
Tacit | Explicit | Implicit

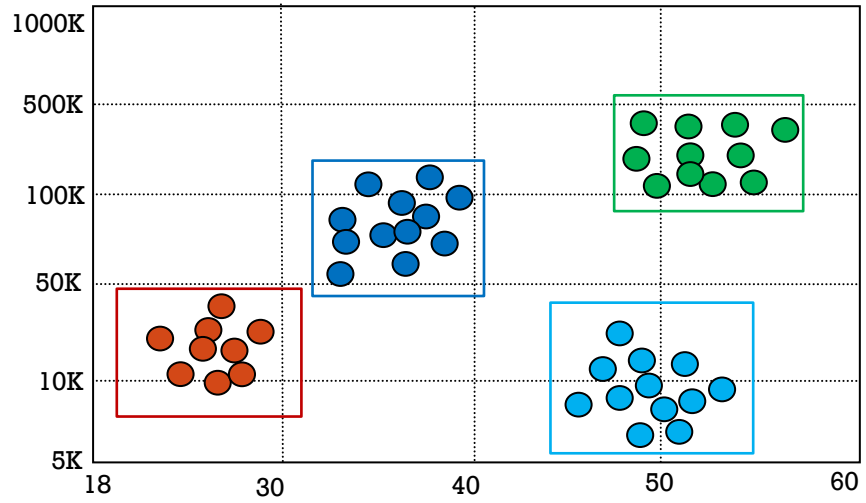
Apply

Think, plan & reason

Problem → Solution

Creating clusters using CBR





1. Decide the matching cutoff based on more niche (higher cutoff) to generalized clusters (lower cutoff). 100% means N clusters 0 means 1 cluster.
2. Mark all cases as un-clustered. Initialize cluster no to 1.
3. Pick up case which is not clustered and run CBR, get matching cases based on cutoff add cases to current cluster no if they are un-clustered or current matching is higher than previous one.
4. Increment cluster number
5. Repeat step 3 and 4 until all cases are clustered.
6. Each cluster will have cases matching with each other to set cutoff or higher.

Example: Clustering using Case-based reasoning

Acquire → Learning

Memory
Adaptability

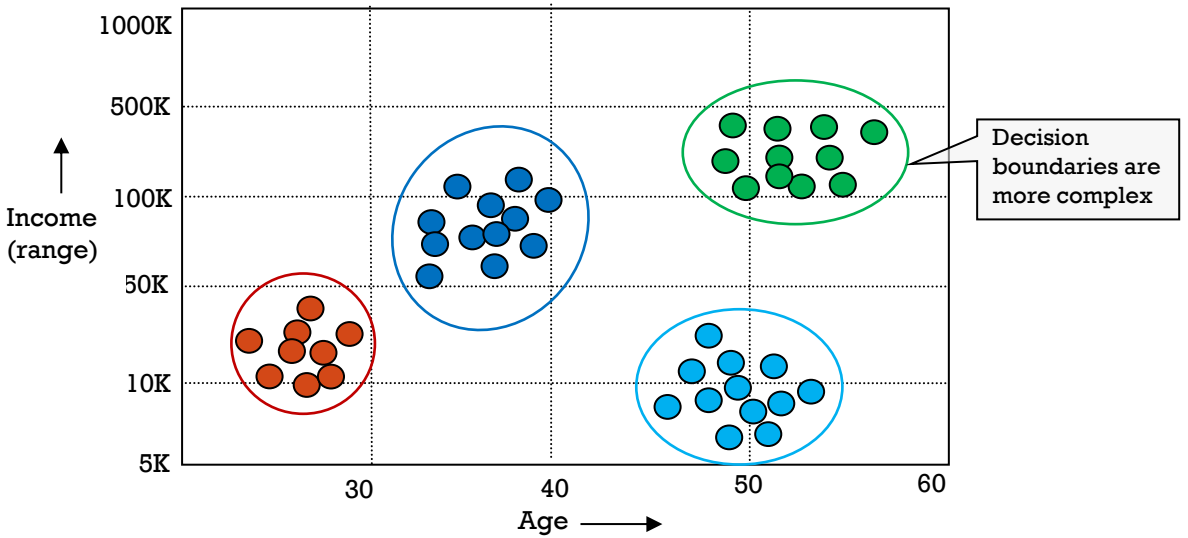
Tacit | Explicit | Implicit

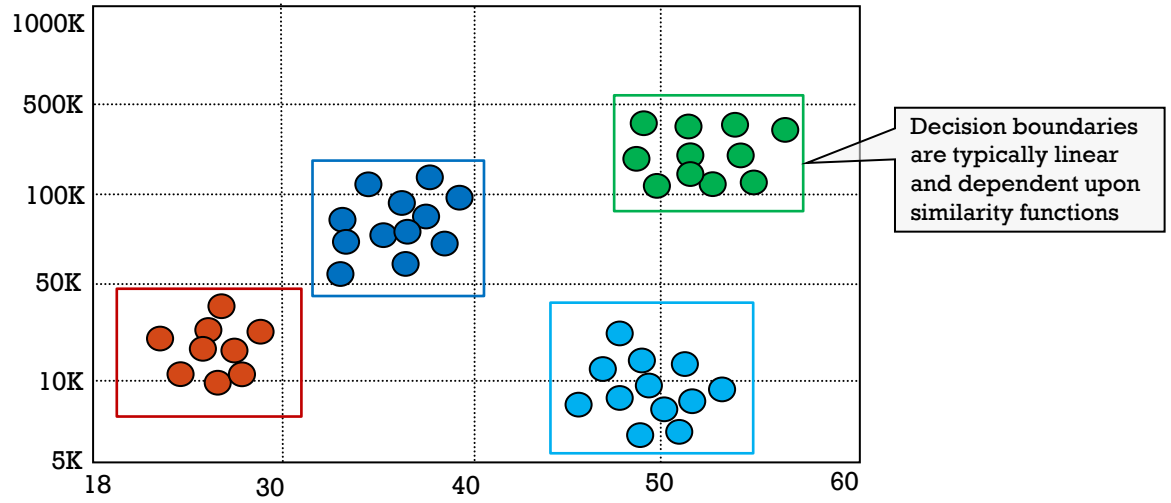
Apply

Think, plan & reason

Problem → Solution

Clustering using ML v/s CBR





Predictive Intelligence using CBR

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

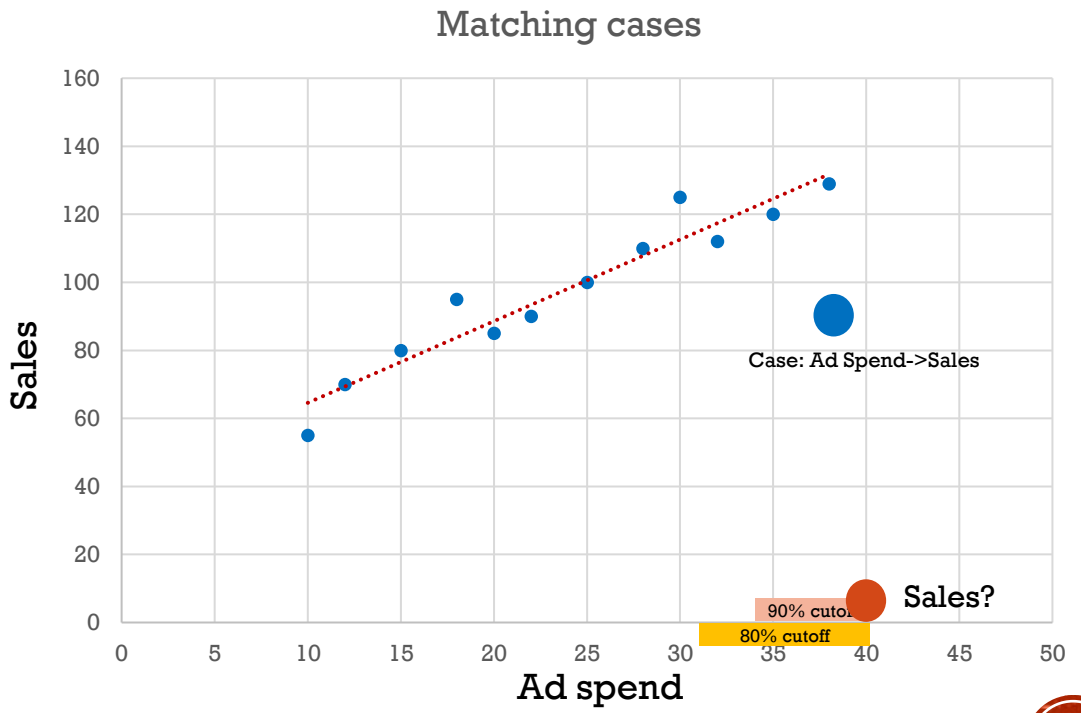
Very oversimplified example: predicting sales using CBR

No	Ad Spend	Sales
1	10	55
2	12	70
3	15	80
4	18	95
5	20	85
6	22	90
7	25	100
8	28	110
9	30	125
10	32	112
11	35	120
12	38	129
13	40	?

Past cases

13	40	?

Problem case: Ad Spend=40



Predictive Intelligence using CBR

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

No	Ad Spend	Sales
1	10	55
2	12	70
3	15	80
4	18	95
5	20	85
6	22	90
7	25	100
8	28	110
9	30	125
10	32	112
11	35	120
12	38	129
13	40	135

Past cases

13	40	?

Problem case: Ad Spend=40

Retrieval: Most similar solved cases: using formula: 1-abs(40- Ad spend)/50 with cutoff 80%

No	Ad Spend	Sales	Matching (%)	Growth (%)
9	30	125	80	13.00
10	32	112	84	-1.04
11	35	120	90	7.10
12	38	129	96	7.50

Average growth rate is 4.4% (because of outlier) else 7%

Reuse: Option 1: Applying 4.4% growth rate sales will be 135 (129*1.44)

13	40	135

Reuse: Option 2: Applying cutoff of 90% average growth rate for no 11 and 12 is 7.32 and sales will be 138 (129*1.732)

13	40	138

Question: compare CBR approach with regression (mathematical)

Think over option 1 and option 2!

Example: CBR based recommendations

Acquire → Learning

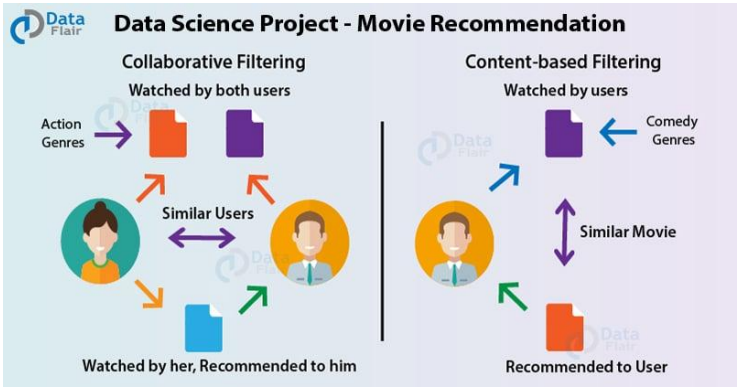
Memory
Adaptability

Tacit | Explicit | Implicit

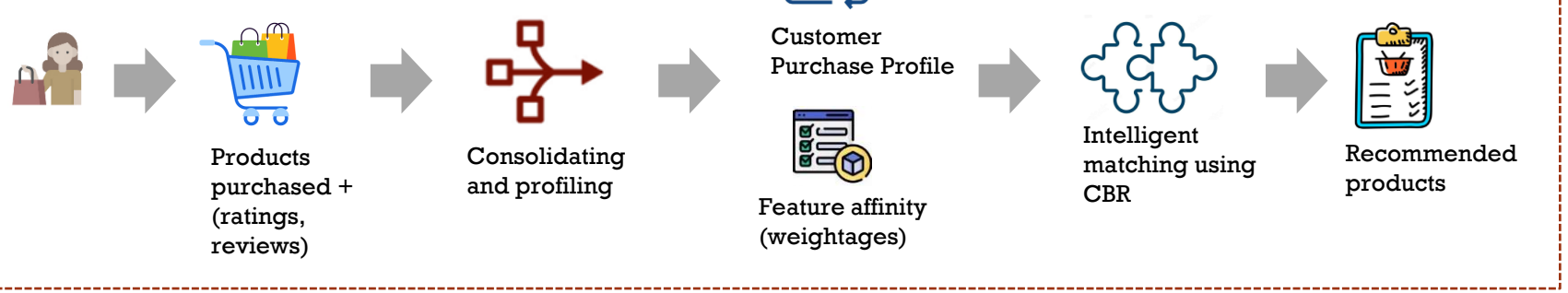
Apply

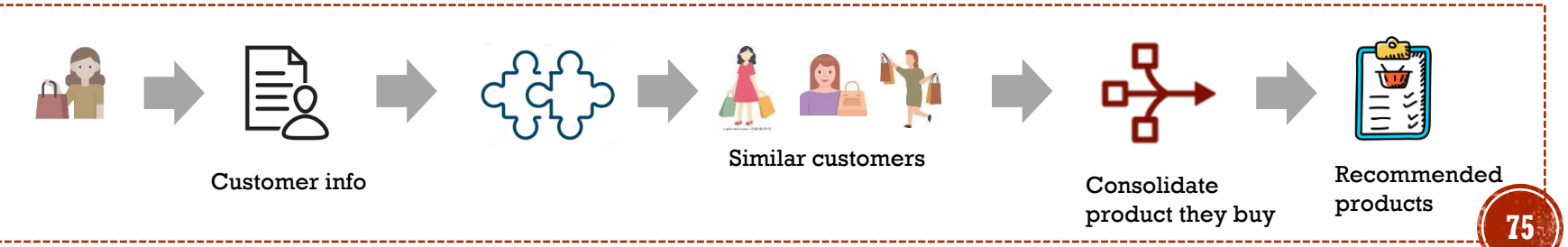
Think, plan & reason

Problem → Solution



1. Collaborative filtering recommends based similar user rating pattern. Does not go into deeper what makes customer buy and rate a product, what are interests the customers? affinity towards brands, colour etc., what are likes and dislikes etc. Has cold start problem.
2. Content filtering typically uses TF-IDF (Term Frequency-Inverse Document Frequency) to find out similar products to products customer likes. Does not go into deeper of semantics of each and every feature, does not understand what is affinity of customer towards a particular feature? Like brand, price, etc.
3. Matching products using CBR based approach can help in very precise recommendation based on very similar products the customer would like. Combined with deep profiling algorithm which determines relative importance of features for every customer can help to provide hyper personalization.





CBR Applications

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

CASE-BASED REASONING

Reuse past experience to solve new problems and improve over time

1. HEALTHCARE & MEDICAL DIAGNOSIS

Helps doctors diagnose rare or complex diseases by comparing with similar patient cases.

Example: Suggest treatment for a patient based on similar past patient histories.

2. TECHNICAL SUPPORT & TROUBLESHOOTING

Resolves technical issues by reusing solutions that worked for similar problems in the past.

Example: Fix a software error by adapting the solution of a previously solved issue.

3. LEGAL & LAW

Assists legal professionals by retrieving similar past cases and their outcomes.

Example: Predict likely judgment or suggest arguments based on similar legal cases.

4. E-COMMERCE & RECOMMENDATION

Recommends products or services by finding similar customer situations and preferences.

Example: Recommend a product based on what worked well for similar customers.

5. FINANCE & BANKING

Supports decision making by comparing with similar financial cases and outcomes.

Example: Assess credit risk or fraud by comparing with similar past transactions.

6. EDUCATION & TRAINING

Provides personalized learning support by adapting solutions or explanations used for similar learner problems.

Example: Suggest learning resources based on how similar students overcame difficulties.

7. ENGINEERING & DESIGN

Solves design or configuration problems by reusing solutions of similar past designs.

Example: Design a machine part by adapting a previous successful design.

8. PUBLIC SERVICES & ADMINISTRATION

Helps authorities make consistent decisions by referencing similar past cases.

Example: Evaluate applications or claims by comparing with similar past decisions.

WHY CBR?

- Leverages real-world experience and knowledge
- Works well in domains where explicit rules are hard to define
- Provides understandable and justifiable solutions
- Continuously improves as new cases are added

Example: Genetic algorithms

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem

Solution

Example: Genetic algorithms

Genetic algorithm should come with equation Y=MX+C for the data given

Acquire → Learning

Memory
Adaptability

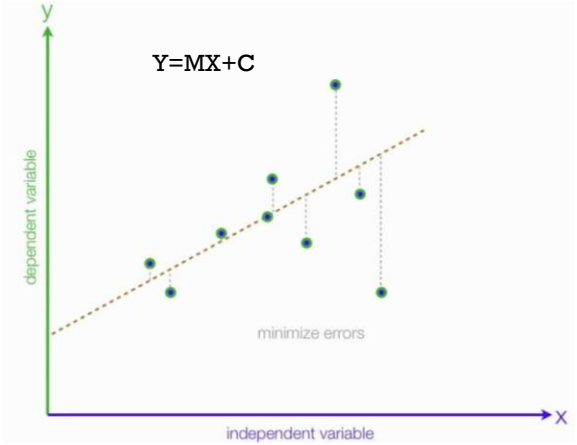
Tacit | Explicit | Implicit

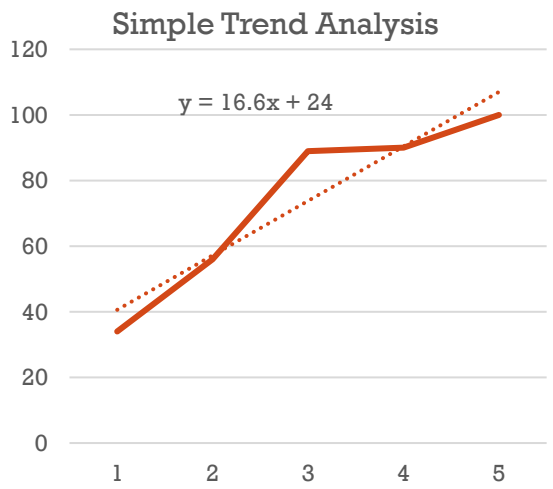
Apply

Think, plan & reason

Problem

Solution





Variables: M & C

Encoding: M & C

Search space: how many possible combinations of values variables

Fitness Function: Average absolute difference between actual and calculated values

Limit search space (possible values of M & C)

Random Values of M & C

Bound values: M & C

Guess values: M & C

Solution No	M	C	Fitness
1	12	12	31.71
2	15	20	12.00
3	16	24	6.57
4	5	23	48.71
5	12	25	19.57

X	Y	Sol 1=12x+12	Abs Diff	Sol 2=15x+20	Abs Diff
1	34	24	10	35	1
2	56	36	20	50	6
3	89	48	41	65	24
4	90	60	30	80	10
5	100	72	28	95	5
6	123	84	39	110	13
7	150	96	54	125	25
			31.71		**12**

Example: Genetic algorithms: Applications

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Genetic Algorithms (GAs) are search and optimization techniques inspired by natural selection and genetics.

1. ENGINEERING DESIGN OPTIMIZATION

Optimize the design parameters of mechanical, electrical, or civil engineering systems.

Example: Minimizing weight of a structure while maximizing strength.

2. SCHEDULING AND TIMETABLING

Find optimal schedules for jobs, machines, employees, or tasks.

Example: Employee shift scheduling, exam timetabling.

GENETIC ALGORITHM

1. Initialize Population
2. Evaluate Fitness
3. Selection
4. Crossover
5. Mutation
6. Repeat until stop condition

3. ROUTE AND PATH OPTIMIZATION

Find the shortest, fastest, or most cost-effective route in logistics and transport.

Example: Traveling Salesman Problem (TSP), vehicle routing.

4. MACHINE LEARNING HYPERPARAMETER TUNING

Automatically search for the best set of hyperparameters for ML models.

Example: Optimizing learning rate, number of layers, regularization.

5. FEATURE SELECTION

Select the most relevant features from a large dataset to improve model performance and reduce complexity.

Example: Selecting important genes in bioinformatics, or variables in datasets.

6. PORTFOLIO OPTIMIZATION

Find the best combination of assets to maximize return and minimize risk.

Example: Stock portfolio selection, asset allocation.

BENEFITS

- Works with complex, non-linear problems
- No need for gradient information
- Global search capability
- Flexible and easy to adapt

7. CONTROL SYSTEMS

Optimize controller parameters for complex systems to improve performance and stability.

Example: PID controller tuning, robotics motion control.

8. BIOINFORMATICS AND DRUG DISCOVERY

Solve complex biological problems like protein structure prediction, gene selection, and drug target identification.

Example: Designing molecules with desired properties.

HOW IT WORKS (SIMPLIFIED FLOW)

Initial Population → Evaluate Fitness → Selection (Best Individuals) → Crossover (Combine) → Mutation (Random Change) → New Population → Repeat until Optimal Solution

Example: Genetic algorithms: Applications

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

1. ENGINEERING DESIGN OPTIMIZATION

- Optimize the design of structures, components or systems
- **Example:** Optimize the shape of an airfoil to maximize lift and minimize drag

2. SCHEDULING & TIMETABLING

- Create optimized schedules under constraints
- **Example:** Job shop scheduling, exam timetabling, employee rostering

8. GAMING & AI

- Evolve strategies, behaviors and game levels
- **Example:** Evolve opponents, strategy optimization in games

3. ROUTING & LOGISTICS

- Find optimal routes and reduce transportation cost/time
- **Example:** Vehicle Routing Problem (VRP), delivery optimization

7. BIOINFORMATICS & HEALTHCARE

- Solve sequence alignment, gene selection, protein folding and drug design problems
- **Example:** DNA sequence analysis, drug discovery

APPLICATIONS OF GENETIC ALGORITHMS

Nature-inspired optimization technique for solving complex search and optimization problems.

4. MACHINE LEARNING & MODEL OPTIMIZATION

- Optimize parameters, feature selection or model structure
- **Example:** Neural network weights, SVM parameters, rule selection

6. FINANCE

- Optimize investment portfolios and trading strategies
- **Example:** Portfolio optimization, risk management, option pricing

5. ADAPTIVE SYSTEMS & CONTROL

- Tune controllers and adapt to dynamic environments
- **Example:** PID controller tuning, robot path planning

KEY IDEA: Genetic Algorithms mimic natural evolution (selection, crossover, mutation) to evolve better solutions over generations.

Genetic Cars

Example: Model-based reasoning

Acquire → Learning

Memory
Adaptability

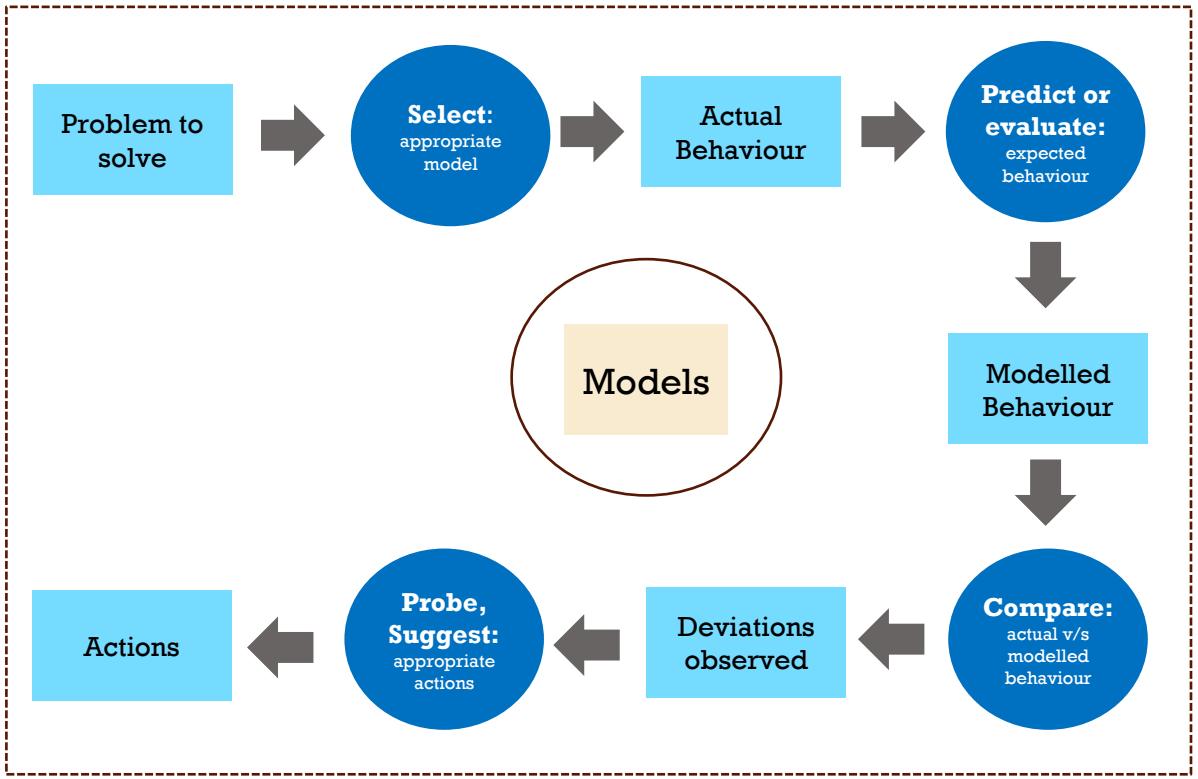
Tacit | Explicit | Implicit

Apply

Problem

Think, plan & reason

Solution



- ✓ Models are widely used to represent real world systems (qualitative as well as quantitative)
- ✓ Models can be static or dynamic models. Dynamic models can be built for specific entities (e.g. customer spending model)
- ✓ Building model-based reasoning system is relatively easier if model is already available. Models can be built at N=1 level to understand behaviours of individual entities like customer risk or opportunity
- ✓ Can address applications like: diagnostics, early warning signals, fault analysis, auditing, repair, troubleshooting

Example: Model-based reasoning

Acquire → Learning

Memory
Adaptability

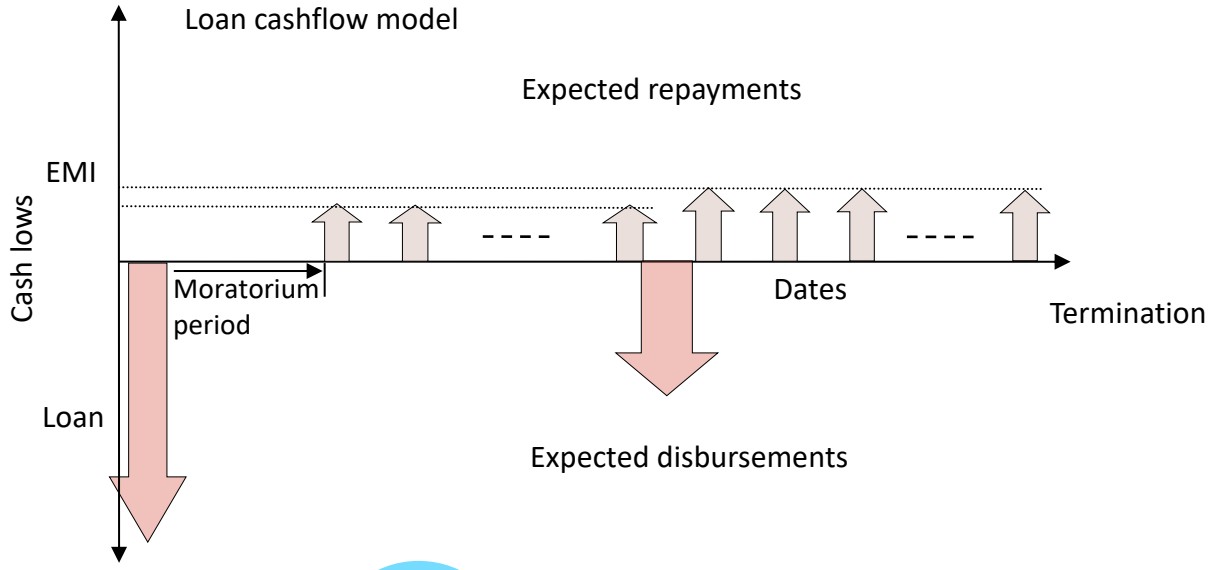
Tacit | Explicit | Implicit

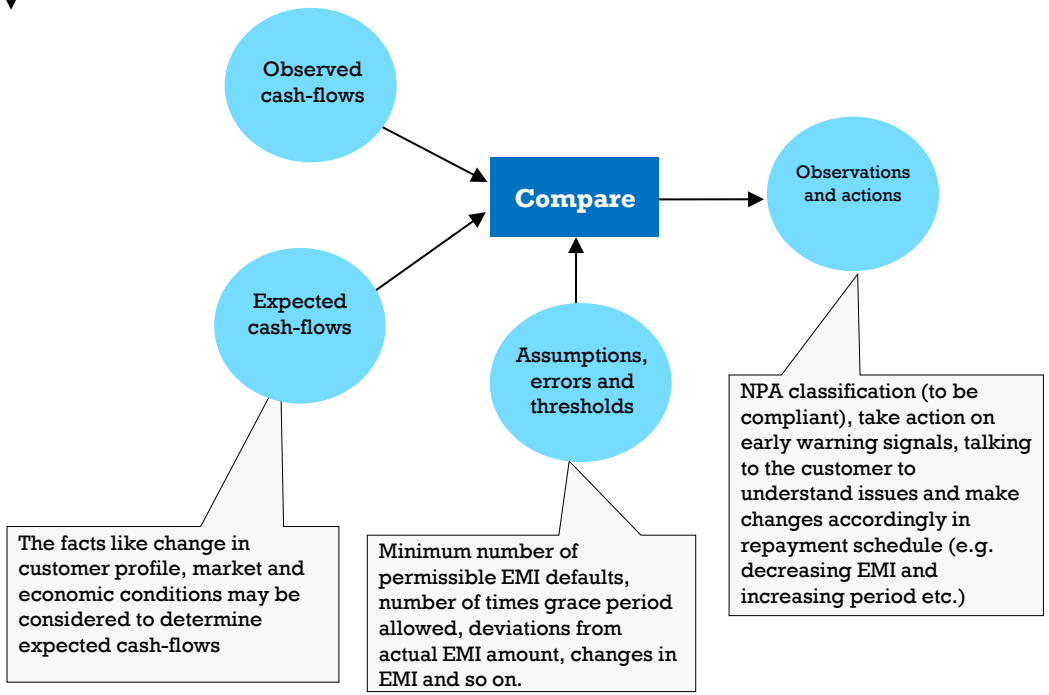
Apply

Think, plan & reason

Problem

Solution





Example: Model-based reasoning

Acquire → Learning

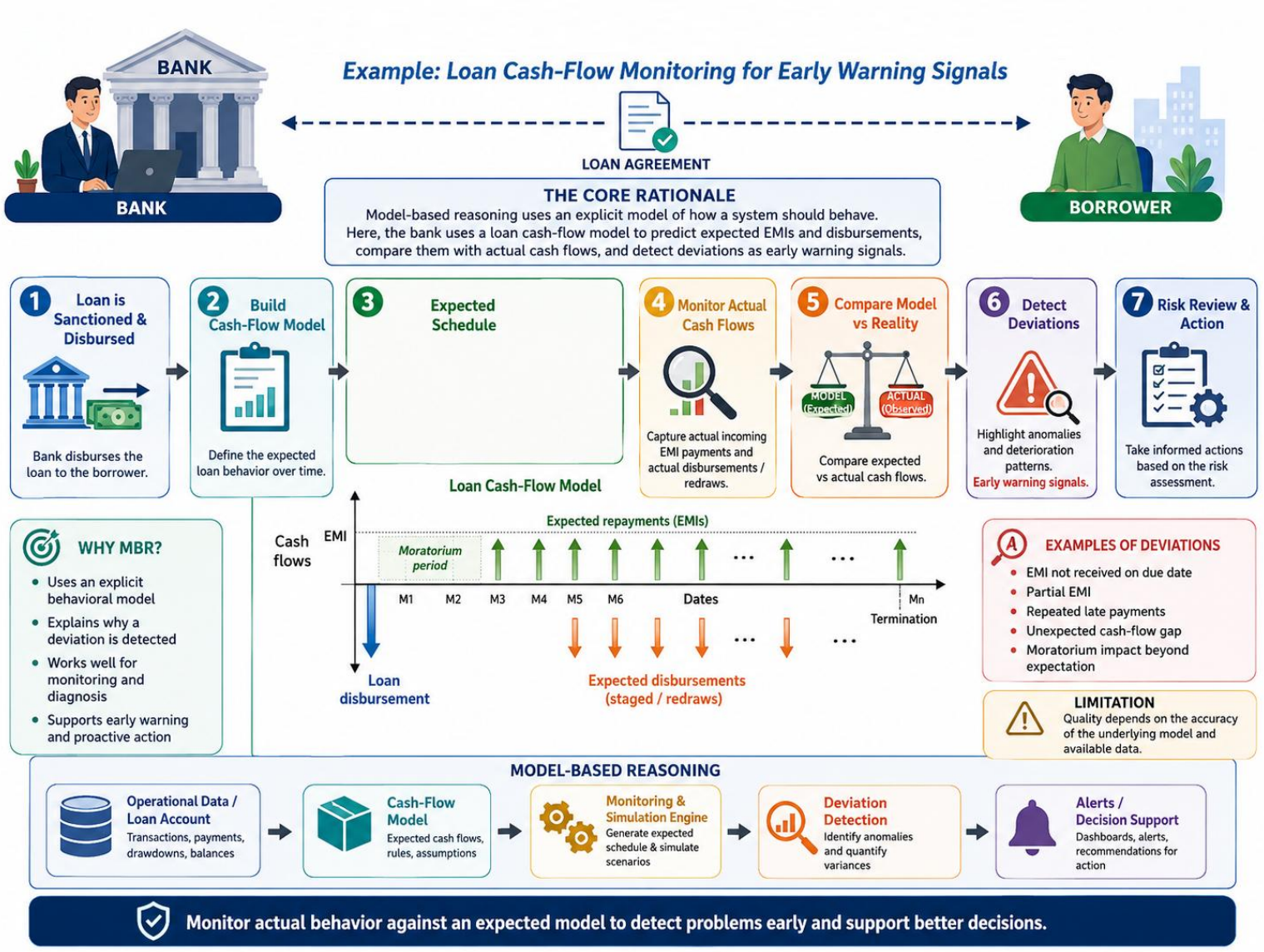
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Example: Model-based reasoning

Acquire → Learning

Memory
Adaptability

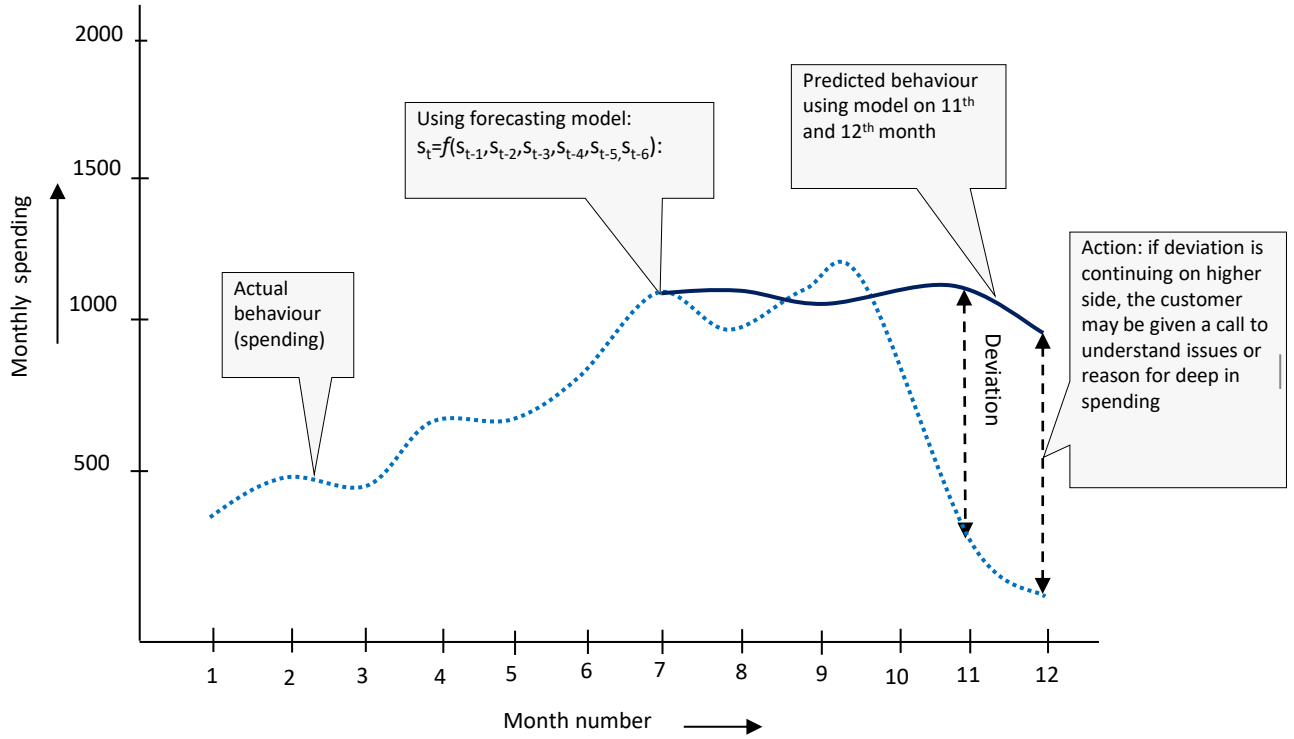
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Customer spending model (dynamic)



Knowledge driven v/s data driven systems

Acquire → Learning

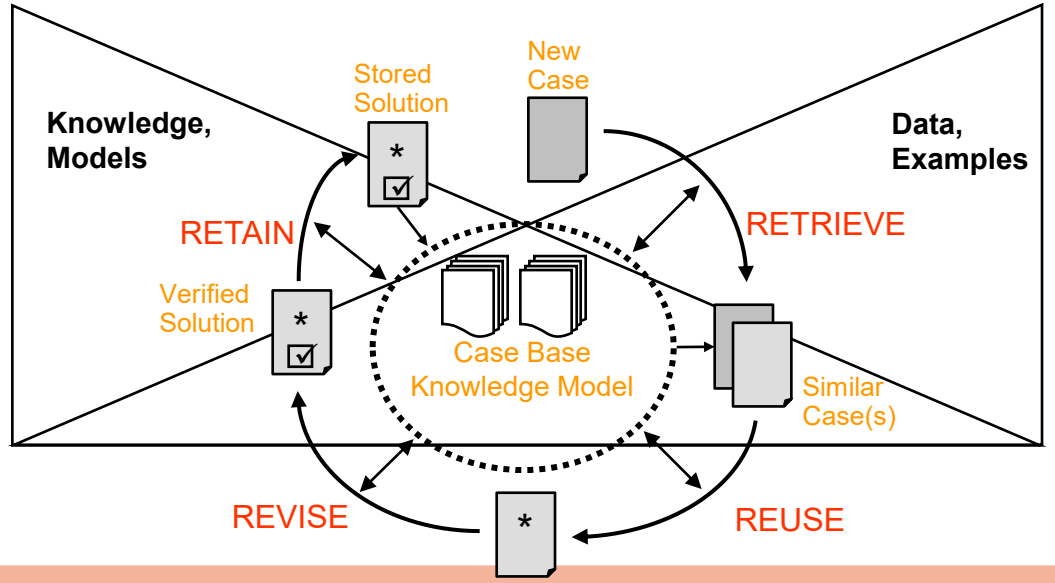
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Closed Knowledge Model — Open Database

Increasing Knowledge Centralization

Increasing Example Orientation

Expert Systems
(completely knowledge driven)

CBR Systems
(knowledge + data driven)

Databases | ML
(completely data driven)

Comparing capabilities: ES, CBR, GA, MBR

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Capability Parameter	Rule-Based Expert System	Case-Based Reasoning (CBR)	Genetic Algorithms (GA)	Model-Based Reasoning (MBR)
Core Concept	Uses IF-THEN rules and inference engine	Solves new problems using past cases	Evolutionary search using selection, crossover, mutation	Uses structural/causal model of the system
Learning Ability	Very Low (manual rule updates)	Moderate (learns by adding new cases)	Adaptive through generations (indirect learning)	Low to Moderate (depends on model updates)
Adaptability to New Situations	Low unless new rules are added	High if similar past cases exist	High in optimization contexts	Moderate (if model is well defined)
Scope to Incorporate Domain Knowledge	Very High (explicit rule encoding)	High (cases + domain annotations)	Low (knowledge not explicitly embedded)	Very High (deep domain modeling)
Knowledge Representation	Explicit rules (symbolic)	Case library + similarity metrics	Chromosomes/fitness functions	System models, constraints, relationships
Explainability	Excellent (transparent reasoning)	High (can show similar past cases)	Low (black-box optimization process)	Very High (causal reasoning traceable)
Dynamism (Real-time Decision Logic)	High (can respond instantly via inference engine)	High (retrieval + adaptation is fast)	Low (iterative and computationally heavy)	Moderate (depends on model complexity)
Data Dependency	Low (knowledge-driven)	Medium (needs historical cases)	Medium (needs fitness evaluation data)	Low to Medium (model-driven)

ChatGpt Generated

Comparing capabilities: ES, CBR, GA, MBR

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Capability Parameter	Rule-Based Expert System	Case-Based Reasoning (CBR)	Genetic Algorithms (GA)	Model-Based Reasoning (MBR)
Handling Uncertainty	Moderate (with fuzzy rules or certainty factors)	High (handles vague similarity)	High (explores uncertain solution space)	High (if probabilistic models used)
Maintenance Effort	High (manual rule maintenance)	Moderate (case base management)	Moderate (parameter tuning)	Very High (model maintenance is complex)
Computational Cost	Low	Moderate	High (population-based iterations)	High (complex simulations/models)
Scalability	Moderate (rule explosion issue)	High (large case libraries manageable)	High for optimization problems	Moderate (complex models scale poorly)
Suitability for Structured Domains	Excellent	Very Good	Limited	Excellent
Real-Time Decision Support	Excellent	Very Good	Poor	Good
Knowledge Acquisition Effort	Very High (knowledge engineering required)	High (case collection required)	Moderate (define fitness + encoding)	Very High (deep domain modeling)
Transparency for Audits/Regulation	Excellent	High	Low	Excellent
Best Use Cases	Compliance systems, policy engines, expert advisory	Helpdesk, diagnosis, recommendation	Scheduling, optimization, routing	Fault diagnosis, engineering systems

ChatGpt Generated

Hybrid Intelligent Systems

Many applications involve multiplicity of intelligence tasks such as diagnosis, scheduling, monitoring, optimization. No one technique can address all these tasks. Each technique has limitations and strengths and may have many sub-components. Integrating other techniques takes care of limitations and enhances the strengths. Based on knowledge and data sources a specific techniques can be used. Even same problem can be addressed by multiple techniques to enhance decision making comfort.

Acquire → Learning

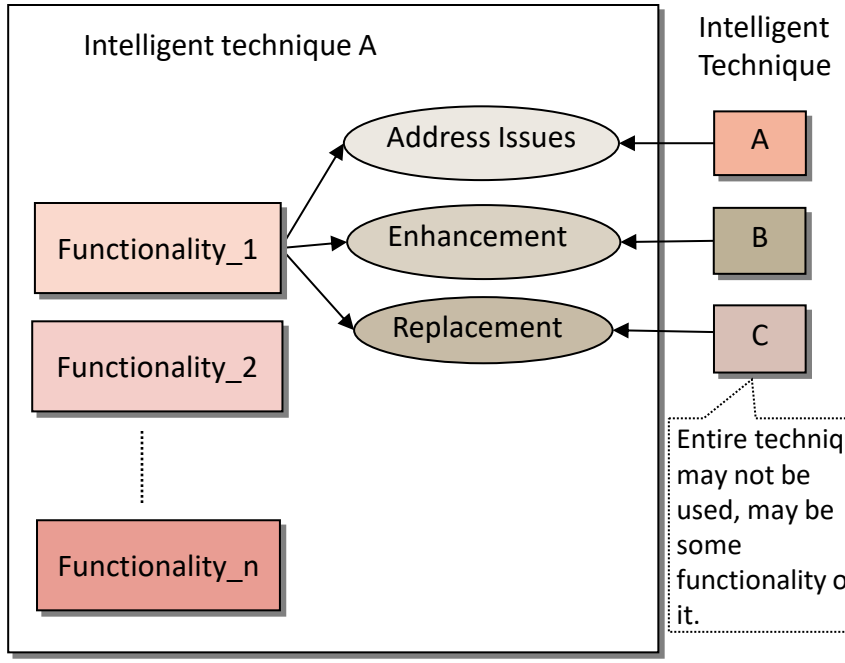
Memory
Adaptability

Tacit | Explicit | Implicit

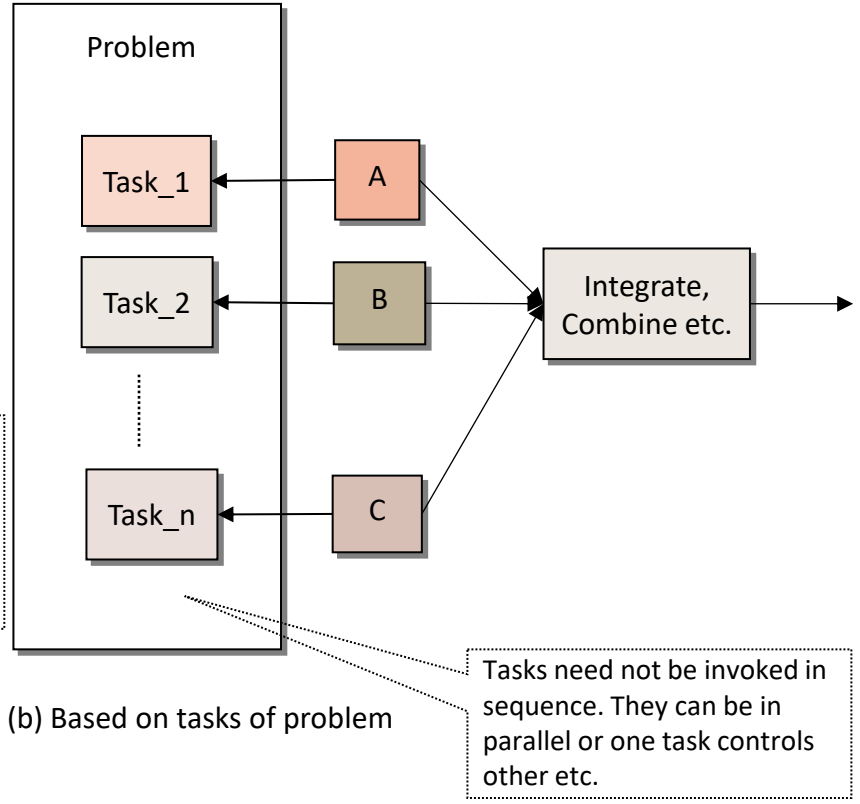
Apply

Think, plan & reason

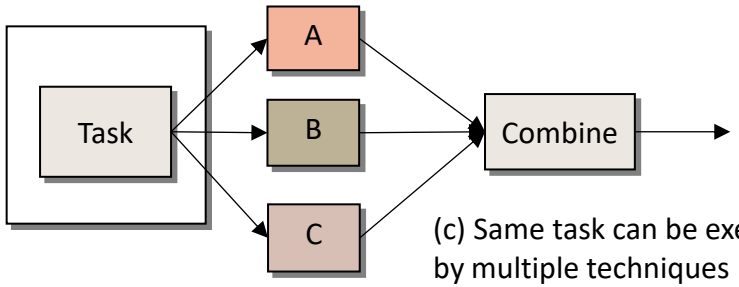
Problem → Solution



(a) Based on functionalities of techniques



(b) Based on tasks of problem



(c) Same task can be executed by multiple techniques

Hybrid Intelligent Systems

Acquire → Learning

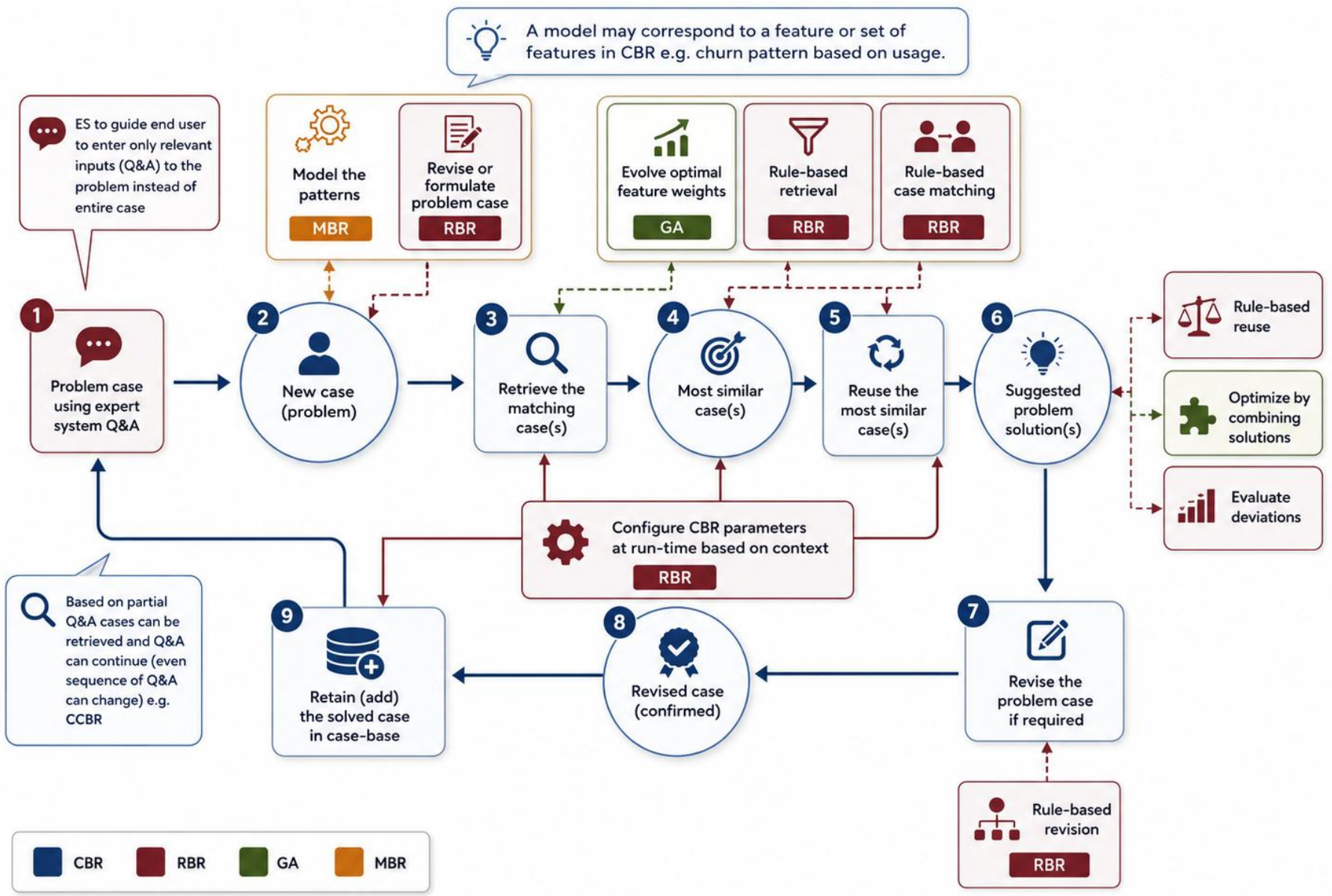
Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution



Example

Hybrid Intelligent Systems: Using effective combinations of technologies

Acquire → Learning

Memory
Adaptability

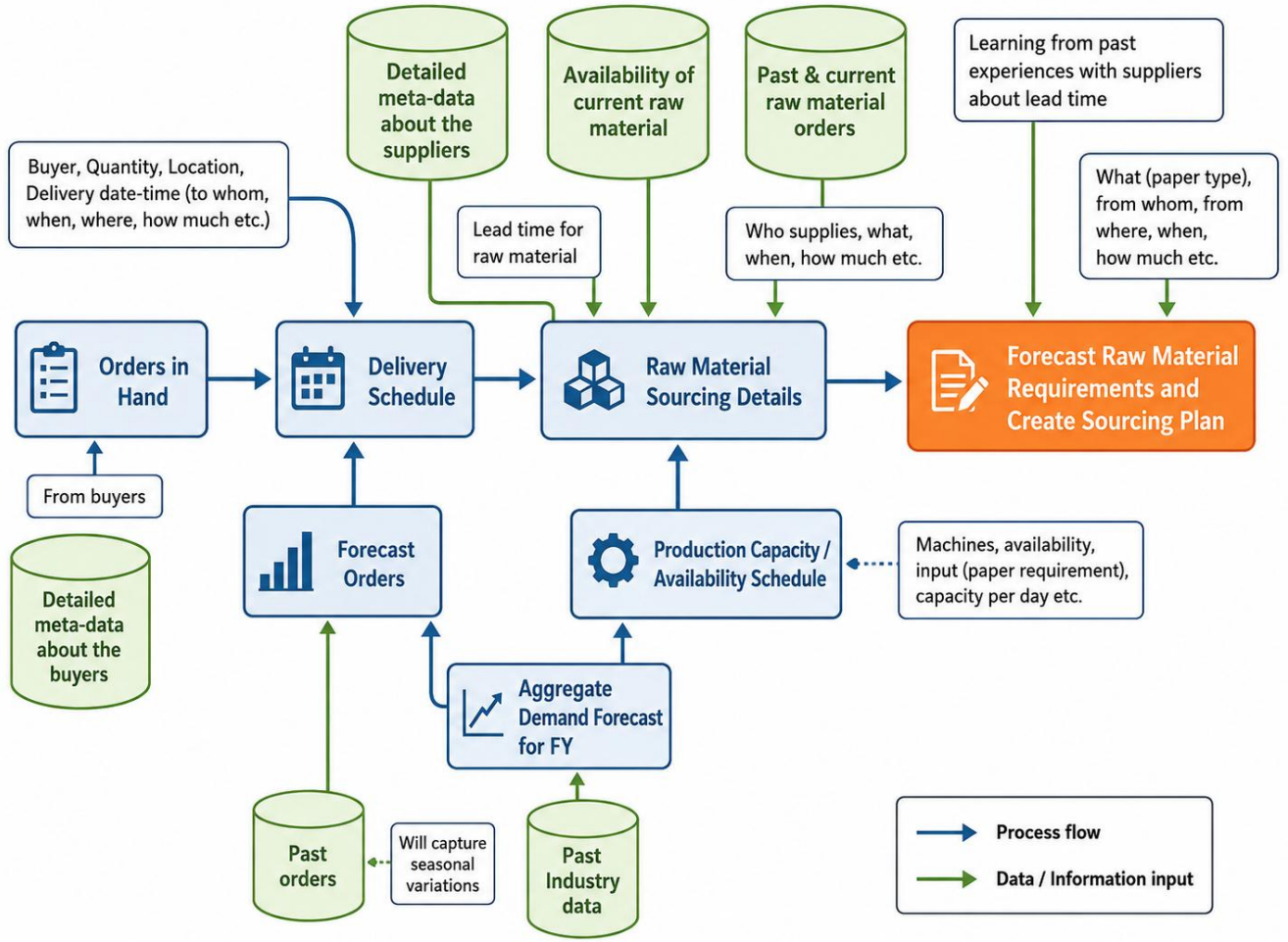
Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Example: A comprehensive intelligent decision support systems to accurately forecast raw material requirements.



Hybrid Intelligent Systems: Using effective combinations of technologies

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Example: A comprehensive intelligent decision support systems to accurately forecast raw material requirements: based on accuracy required, different datasets may be considered.

Data points		
Cardboard meta-data	Details of size, material required, etc.	Required for computations
Raw Material meta-data	Details about type of raw material (e.g. paper, paper type etc.)	
Machines	Output capacity per/hour day Input paper type (grade/size etc.)	To calculate tentative production capacity
People	Details about job skills	
Resource Availability	Availability of resources	
Orders in Hand	Buyer (who), location (at what location), what type of paper, how much, at what time, etc.	Create delivery schedule in detail
Buyers	Details, location	To forecast likely demand from the existing buyers and to certain extent new buyers based on market data
Past Orders	Order date-time, shipment location and date-time, delivery date-time, card-board type, location, etc.	
Industry data	Market reports, market past data, industry forecasts	
Suppliers	Details, location, capacity to supplier paper types, typical lead time (paper-wise), etc.	To calculate/forecast lead-time in order to have lean or optimized inventory
Past Raw material orders	Order date-time, shipment date-time, received date-time, paper type, from supplier, from location, etc.	
Current finished and raw material inventory	Details of current orders and raw material availability	

Hybrid Intelligent Systems: based on data/expertise availability

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

- No user transactions → Ranked items | Similar Items | Common sense Item Intelligence
- Some user transactions → Business rules + Similar items
- Enough transactions available for generalization → Similar items + items by CF + ranked items
- Enough user transactions available → Items based on content filtering + other
- Demographics available → Items based on DF + Ranking + other

Hybrid Intelligent Systems: Integrating various services

Acquire → Learning

Memory
Adaptability

Tacit | Explicit | Implicit

Apply

Think, plan & reason

Problem → Solution

Push service: rating would be sent to N=1 Analytics immediately

My page

Selected item (I_ID)

Notify Transaction
(P_ID, C_ID,I_ID, L_ID, Rating)

Rate now: *****

People also bought

Get Associated Items
(I_ID, C_ID, L_ID, T_ID)

Service that would return list of item IDs that are related to selected item

Exclusive offers for you

Get Preferred Items
(P_ID, C_ID, L_ID, T_ID)

Related items

Get Related Items
(I_ID, C_ID, L_ID, T_ID)

Recommended to you

Get Personalized Items
(P_ID, C_ID, L_ID, T_ID)

Similar Items

Get Similar Items
(P_ID, L_ID, T_ID)

Generic widget where items related to selected item would be populated

Popular items

Get Popular Items
(P_ID, C_ID, L_ID, T_ID)

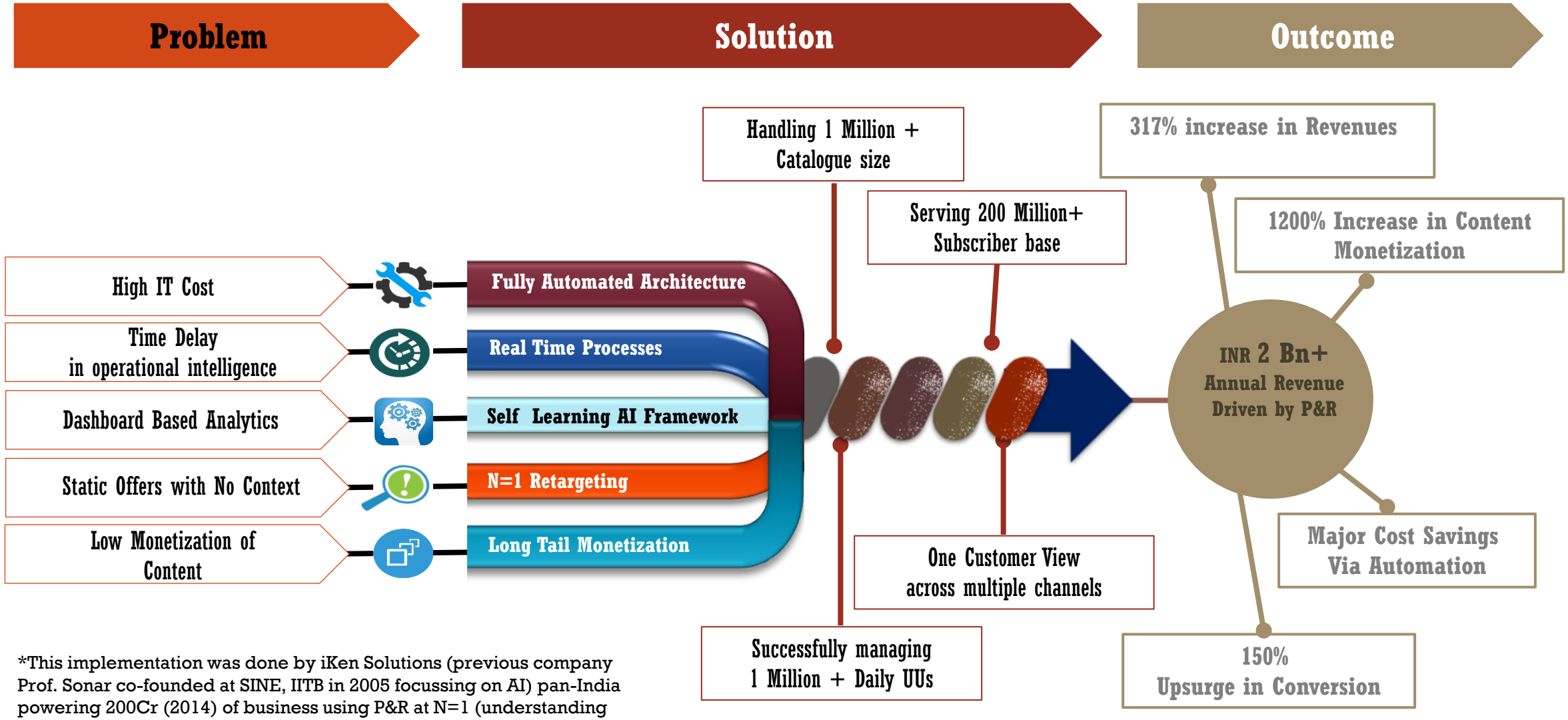
Users like you

Get Related Consumers
(P_ID, L_ID, T_ID)

P_ID: Party ID (User), I_ID: Item ID, L_ID: location ID of party, C_ID: Category ID

Towards Automation of Business Intelligence Services Using Hybrid Intelligent System Approach, 2013

P&R for a telecom value added services*



*This implementation was done by iKen Solutions (previous company Prof. Sonar co-founded at SINE, IITB in 2005 focussing on AI) pan-India powering 200Cr (2014) of business using P&R at N=1 (understanding one entity at a time)

Back

Key takeaways

AI leadership needs fair understanding of what problems AI/ML address, how they work and applications they address

Different problem types
There are many generic problem types exist based on type of fundamental tasks. One generic problem type is understood one can think of how variety of use-cases in practical scenarios in various verticals, functions and domain can be modelled.

Hybrid Systems
No single technique/algorithm solves all problem. Enterprise level problems are complex and need to connect multiple systems complementing their functionalities or improving the overall decision making and leveraging data and human knowledge.

Understand problem solving capabilities of core AI/ML and role of data and knowledge

Knowledge-Driven systems
Domain knowledge is very important to automate knowledge and decision making. These systems can play role in cold-start problems when no data is available, retain expertise as well as building knowledge from Gen AI validated and verified by human experts.

Combining semantic and core AI/ML
Adding semantics on top of or along with other techniques can help to address problem more effectively and accurately. Leverage human and GenAI expertise.