Machine Reasoning
Day 2

Course Manager: GU Zhan (Sam)

zhan.gu@nus.edu.sg

Machine Reasoning

Day 2

DAY 2 AGENDA

2.1 Machine Reasoning Enabler: Knowledge Representation

- 2.1.1 Forms of Knowledge Representation
- 2.1.2 Forms of Knowledge Representation (Rules)
- 2.1.3 Exercise

2.2 Machine Reasoning Enabler: Knowledge Acquisition

- 2.2.1 Knowledge Acquisition (manual elicitation vs auto discovery)
- 2.2.2 Knowledge Elicitation using Knowledge Models
- 2.2.3 Knowledge Discovery using Data Mining Models
- 2.2.4 Exercise

2.3 Knowledge Representation and Acquisition/Discovery [Workshop]

- 2.3.1 Deductive Reasoning (Decision Tree)
- 2.3.2 Inductive Reasoning (Topic Summarization)
- 2.3.3 Analogical Reasoning (FAQ Knowledge Bot)
- 2.3.4 Workshop Submission

2.1 Machine Reasoning Enabler: Knowledge Representation

2.1.1 Forms of Knowledge Representation

2.1.2 Forms of Knowledge Representation (Rules)

2.1.3 Exercise

2.1 Machine Reasoning Enabler: Knowledge Representation

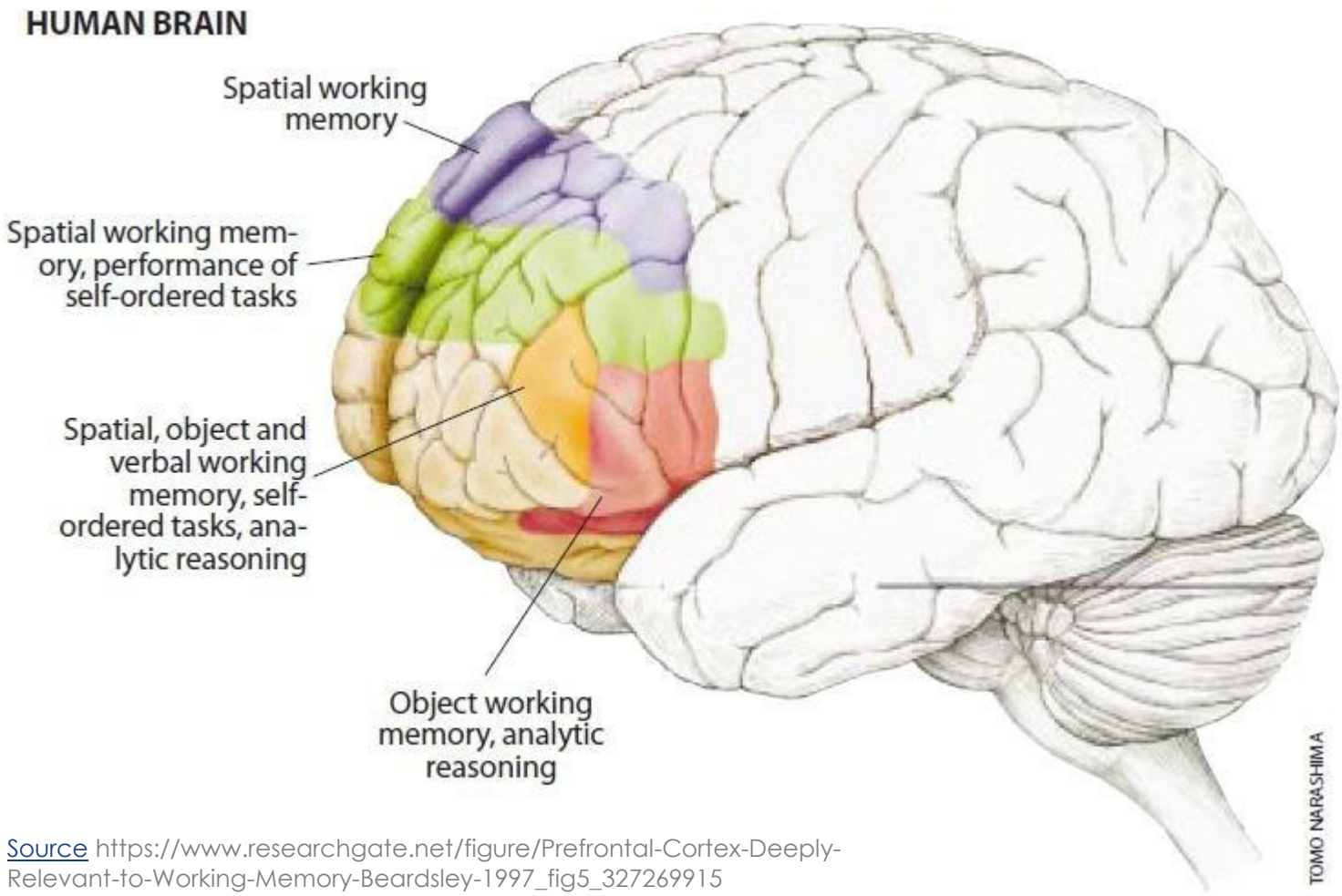
2.1.1 Forms of Knowledge Representation

2.1.2 Forms of Knowledge Representation (Rules)

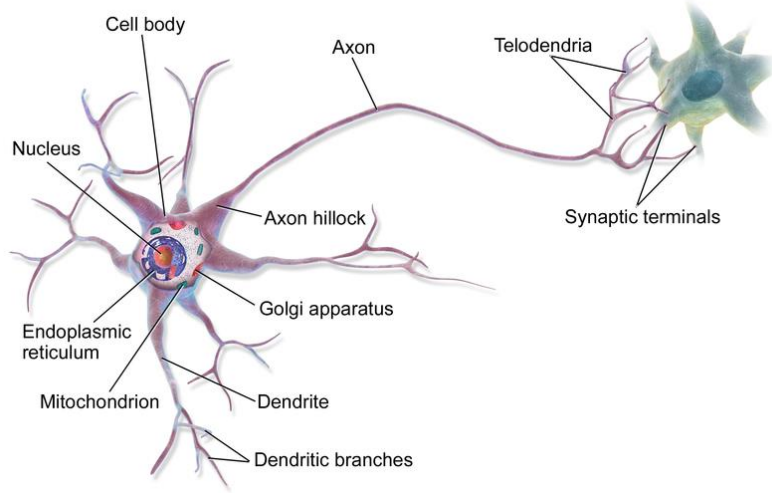
2.1.3 Exercise

Forms of Knowledge Representation

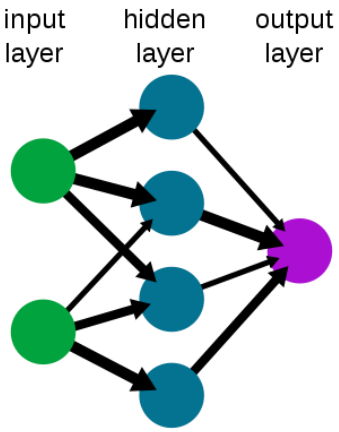
- **Knowledge representation in human brain** (black box)



Source https://www.researchgate.net/figure/Prefrontal-Cortex-Deeply-Relevant-to-Working-Memory-Beardsley-1997_fig5_327269915



A simple neural network



Forms of Knowledge Representation

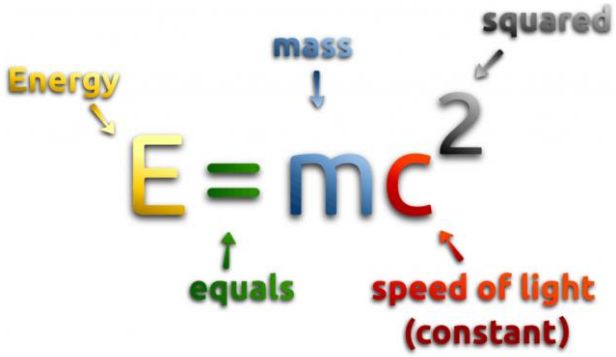
- **Knowledge representation in machine** (white box)**: A scheme /method that allows the computer system to use or manipulate it to reason and solve problems**
  - Unlike humans, knowledge must be 'transplanted/saved' into machine reasoning system/memory
  - Representation goes hand-in-hand with reasoning/inference mechanism (computer algorithm)
  - Large amount of knowledge is usually needed to solve complex problems
- **Explicit knowledge representation (think of documentation) enables business knowledge management and retention.**

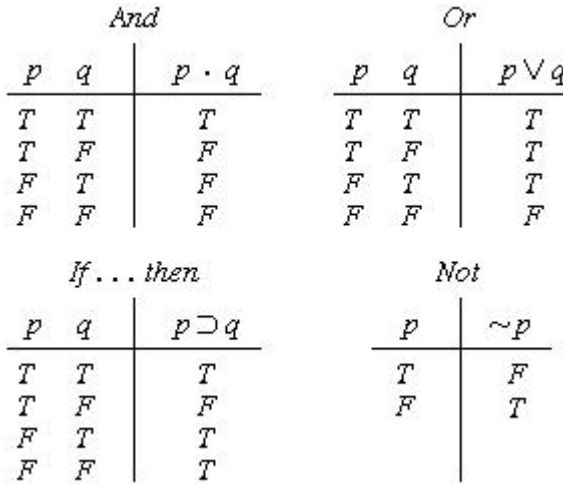
Data Structure

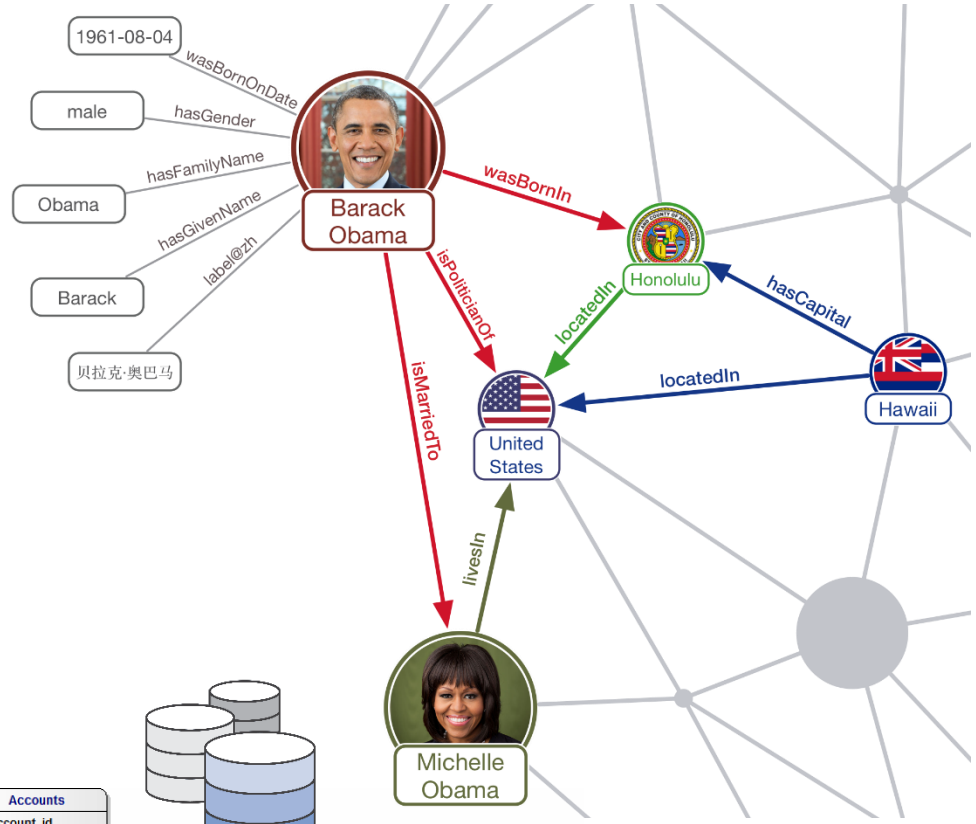
Processing Logic

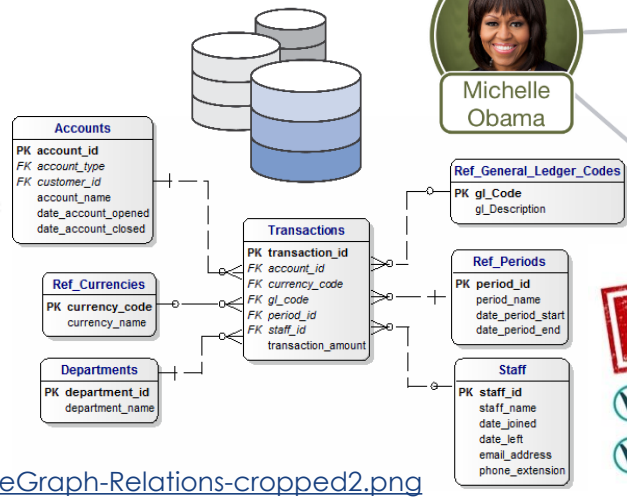
Forms of Knowledge Representation

- Natural Language (Text)
- Formula
- Formal Logic
- Semantic Web
- Frames
- Ontology
- Knowledge Graph
- Database
- Rules
- And many other forms…









https://www.ambiverse.com/wp-content/uploads/2017/01/KnowledgeGraph-Relations-cropped2.png

Forms of Knowledge Representation

Formal Logic – Propositional Logic

- **Propositional Logic**
  - Examples of propositions: propositional sentence
    - ***p*** = “Sam has flu.”
    - ***q*** = “Sam has cough.”

And

p	q	$p \cdot q$
T	T	T
T	F	F
F	T	F
F	F	F

Or

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

If . . . then

p	q	$p \supset q$
T	T	T
T	F	F
F	T	T
F	F	T

Not

p	$\sim p$
T	F
F	T

  - What about sentence: ***s*** = “x + y = 5”, where x and y are variables?

    ☺ Not a proposition, as its truth cannot be defined unless x and y are assigned specific values

Forms of Knowledge Representation

Formal Logic – Propositional Logic

- **The syntax of propositional logic expression is constructed using propositions and connectives**
  - All propositions must be either **true** or **false** (referred to as **truth value** of the proposition)
- **Connectives**
  - ¬ negation "not"
  - ∨ disjunction "or"
  - ∧ conjunction "and"
  - → implication "if … then"
  - ↔ bi-conditional "if and only if"
- **Complex expressions using connectives**
  - $p \wedge q$ = "Sam has flu. **AND** Sam has cough."
  - $p \wedge \neg q$ = "Sam has flu. **AND** Sam has no cough."
  - $p \rightarrow q$ = "**IF/WHEN** Sam has flu **THEN** Sam has cough."

Forms of Knowledge Representation

Formal Logic – Logical equivalence

- **Logical equivalence**

  Two sentences are logically equivalent is written as: **$\alpha \equiv \beta$** (iff $\alpha \models \beta$ and $\beta \models \alpha$)

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Forms of Knowledge Representation

Formal Logic – First Order Logic

- **First Order Logic (Predicate Calculus)**
  - Propositional logic assumes the world contains: **facts**
  - First order logic (also called first-order predicate calculus, or predicate logic) assumes the world contains:
    - **Objects (Class)** : people, houses, numbers, colors, baseball games, …
    - **Constants (Instance)** : The White House, Sam GU Zhan, π, NUS,...
    - **Variables** : x, y, a, b, ...
    - **Relations (Predicate)** : is student, has leg, eats, is bigger than, is part of, is red, is round, prime to, come between, is one more than, …
    - **Functions (Predicate)** : father of, best friend, square root of, sum of, …
    - **Connectives** : ¬ → ∧ ∨ ↔
    - **Equality** : =
    - **Quantifiers** : ∀ ∃

Forms of Knowledge Representation

Formal Logic – First Order Logic

- **Sentences of First Order Logic**
  - **Term** (noun) is an expression that refers to an object.
    - FatherOf(DiDi) : "(a person, who is) father of DiDi"
    - ¬ FatherOf(DiDi) : "(a person, who is) not father of DiDi"
    - FatherOf(x) : "someone's father"
    - Sam, Jessie, DiDi, Machine Reasoning Course, PhD, …
    - Integer: x, y, z (variables of object: all integer numbers)
  - **Atomic Sentence** (semantics) is formed from **one** predicate symbol followed by **one** parenthesized list of terms

Relational predicate
- IsFriend(Jessie, Sam) : Jessie is friend to Sam.

Functional predicate
- FatherOf(DiDi) : DiDi's father (Result is Sam.)

Embedded
- IsFriend(Jessie, FatherOf(DiDi)) : Jessie is friend to DiDi's father.

Forms of Knowledge Representation

Formal Logic – First Order Logic

- **Sentences of First Order Logic**
  - **Complex Sentence** (semantics) is made from **Atomic Sentences** using logical connectives
    - IsClassmate(Jessie, Sam) $\wedge$ TalksTo(Jessie, Sam) $\rightarrow$ IsFriend(Jessie, Sam)
- **Establish Truth of Predicate**
  - Predicates need to be propositionalized for use in reasoning
  - **Method 1**: Assign specific value to predicate expressions (similar to instantiation)
    - StudyAt(x, NUS) $\rightarrow$ Smart(x) What's the scope of x?
    - x = Sam
    - Conclusion: StudyAt (Sam, NUS) $\rightarrow$ **Smart(Sam)**

Forms of Knowledge Representation

Formal Logic – First Order Logic

- **Establish Truth of Predicate**
  - **Method 2A**: Universal quantifier: **∀**

    We want to express "Everyone studying at NUS is smart."

    √ $\forall x\ StudyAt(x, NUS) \rightarrow Smart(x)$: "For everyone, if the person is studying at NUS then the (**same**) person is smart" (For those are not studying at NUS, we don't know.)

    × $\forall x\ StudyAt(x, NUS) \wedge Smart(x)$: "Everyone (all persons in Singapore) is studying at NUS and all (**these**) persons are smart." **incorrect semantic**
  - **Method 2B**: Existential quantifier: **∃**

    We want to express "Someone studying at NUS is smart."

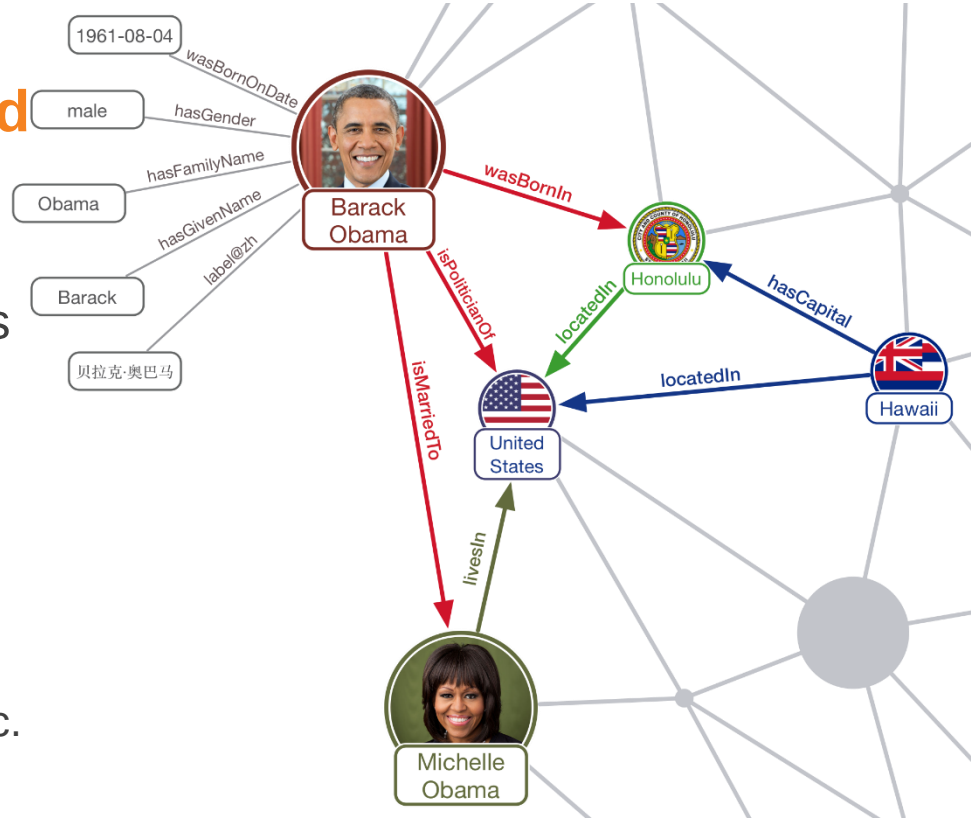
    √ $\exists x\ StudyAt(x, NUS) \wedge Smart(x)$: "There is someone studying at NUS and this (**same**) person is smart."

    × $\exists x\ StudyAt(x, NUS) \rightarrow Smart(x)$: "There is someone, when he/she is studying at NUS then he/she is smart." (When this (**same**) person is having a rest, then he/she may not be smart.) **incorrect semantic**

Forms of Knowledge Representation

Semantic Web, Knowledge Graph

- **Semantic web is a model for word concepts in human cognition, consisting of nodes, links and labels**
  - **Nodes (vertices)** represent objects, concepts, or situations. They can be instances (individual objects as in Knowledge Graph) or classes (generic objects as in Semantic Web)
  - **Links (edges)** between nodes represent a relationship/predicate
  - **Labels (attributes)**
    - Labels on nodes indicate the name of the object, concept, etc.
    - Labels on links describe the type of relationship between nodes
- **Reasoning question: What's the relationship between Barack and Michelle Obama?**

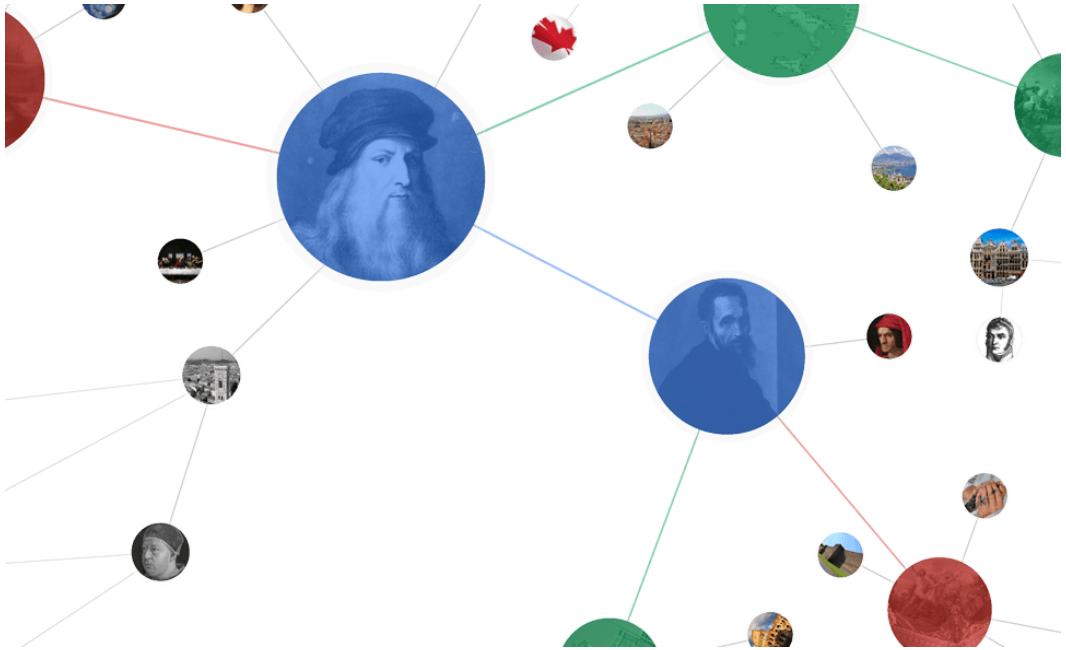


https://www.ambiverse.com/wp-content/uploads/2017/01/KnowledgeGraph-Relations-cropped2.png

Forms of Knowledge Representation

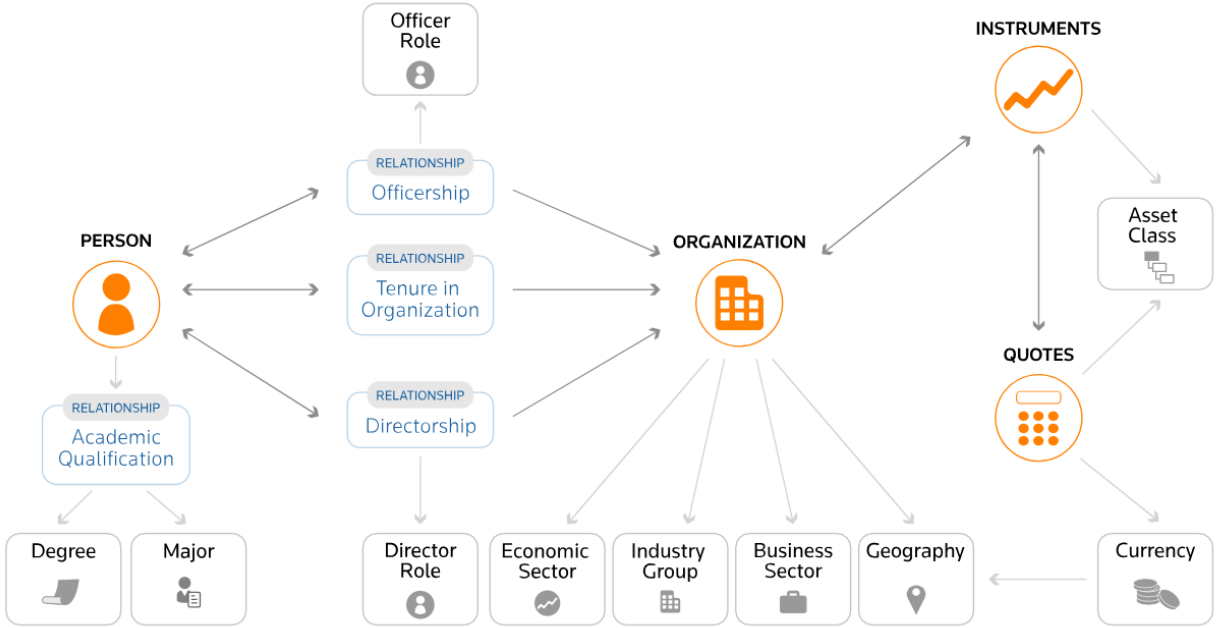
Semantic Web, Knowledge Graph

- **Google Knowledge Graph**



https://www.tampa-seo.com/wp-content/uploads/static-graph.png

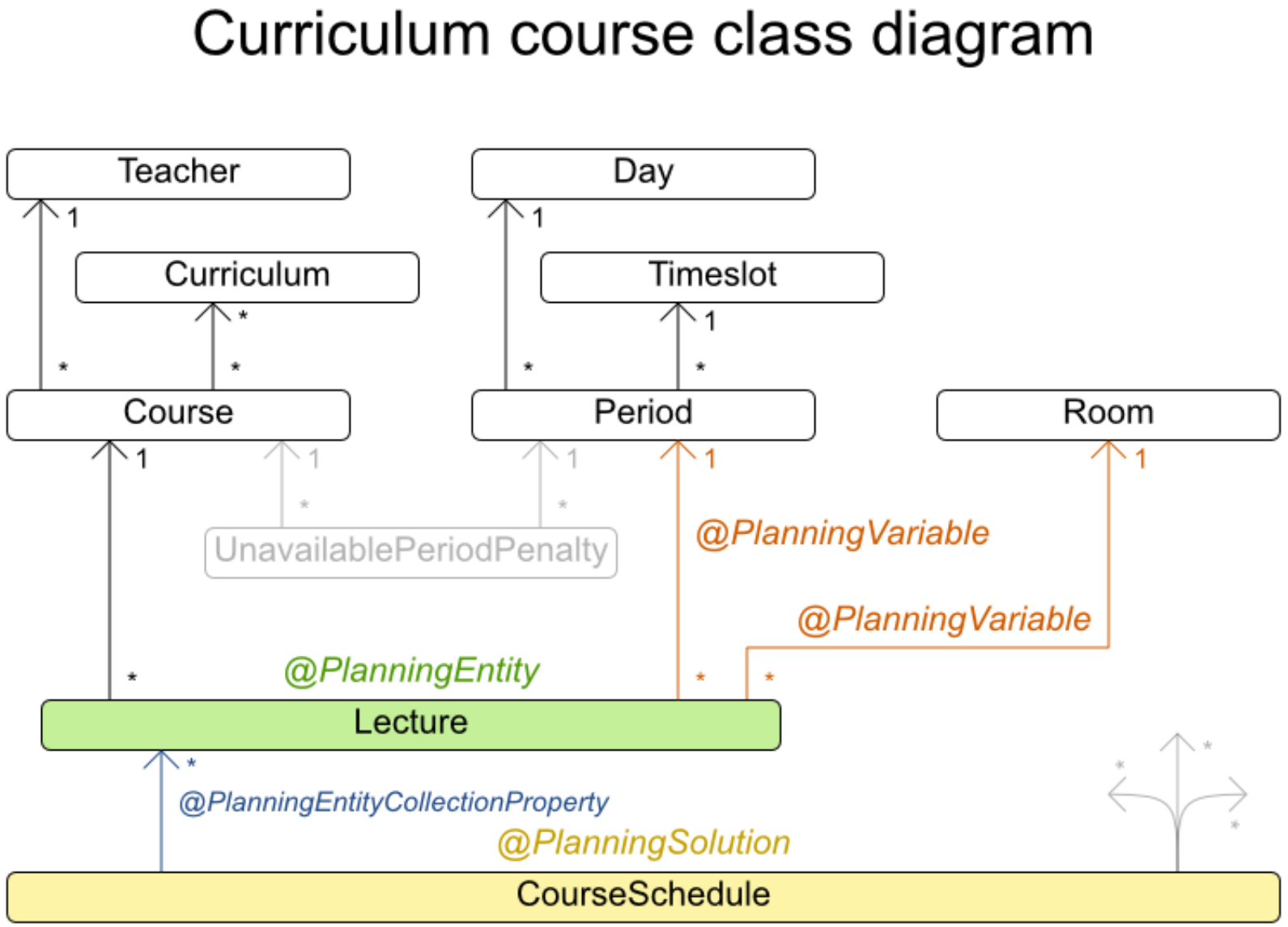
- **Thomson Reuters Knowledge Graph product: Perm ID**



https://permid.org/

Forms of Knowledge Representation

Domain Ontology / OO Classes / DB Schema



2.1 Machine Reasoning Enabler: Knowledge Representation

2.1.1 Forms of Knowledge Representation

2.1.2 Forms of Knowledge Representation (Rules)

2.1.3 Exercise

Forms of Knowledge Representation (Rules)

Rules (A form of logic: propositional or first order)

- **Represent problem-solving knowledge as**

  "**IF/WHEN** … **THEN** …" rules

  - Is the classic technique for representing domain knowledge in an machine reasoning system
  - Is also a very natural way of human decision making

- **A rule consists of two parts:**
  - The **IF** part
    - called the **antecedent** or **premise** or **condition**
  - The **THEN** part
    - called the **consequent** or **conclusion** or **action**

Forms of Knowledge Representation (Rules)

Rules

- **Basic Rule Syntax**

IF <antecedent>

THEN <consequent>

IF person X is ill

THEN person X need rest a lot

Forms of Knowledge Representation (Rules)

Rules

- **Multi-antecedent Rule**

IF <antecedent 1>
AND/OR <antecedent 2>
…
AND/OR <antecedent n>
THEN <consequent>

IF person X is ill
AND person X is a lecturer
THEN person X cannot rest at home, but go to class

Forms of Knowledge Representation (Rules)

Rules

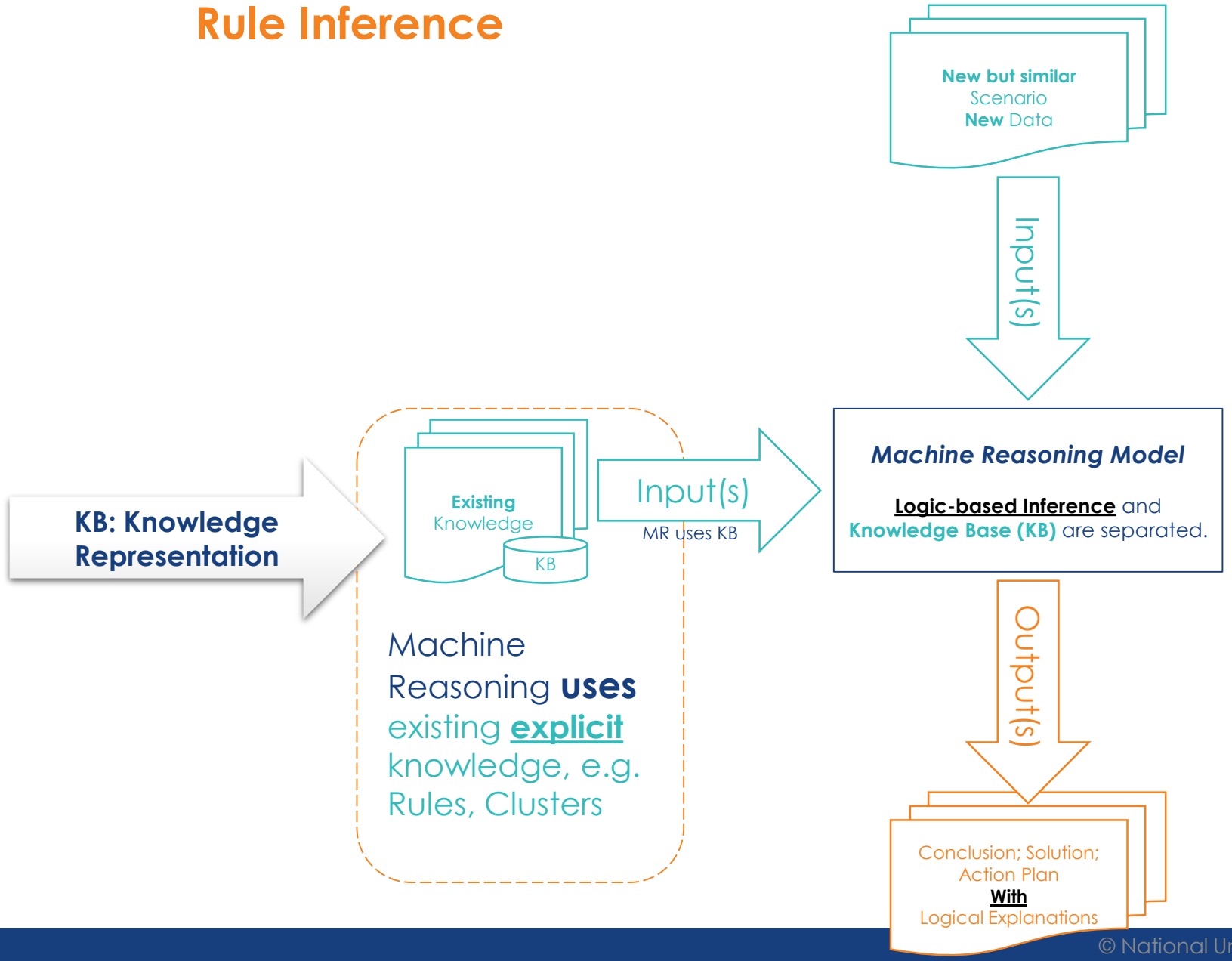
- **Multi-consequent Rule**

  IF <antecedent 1>
  THEN <consequent 1>
  <consequent 2>
  …
  <consequent m>

☺ **The relationship between the multiple consequents is understood as AND, depending on the implementation of software for developing reasoning systems.**

Forms of Knowledge Representation (Rules)

Rule Inference



Forms of Knowledge Representation (Rules)

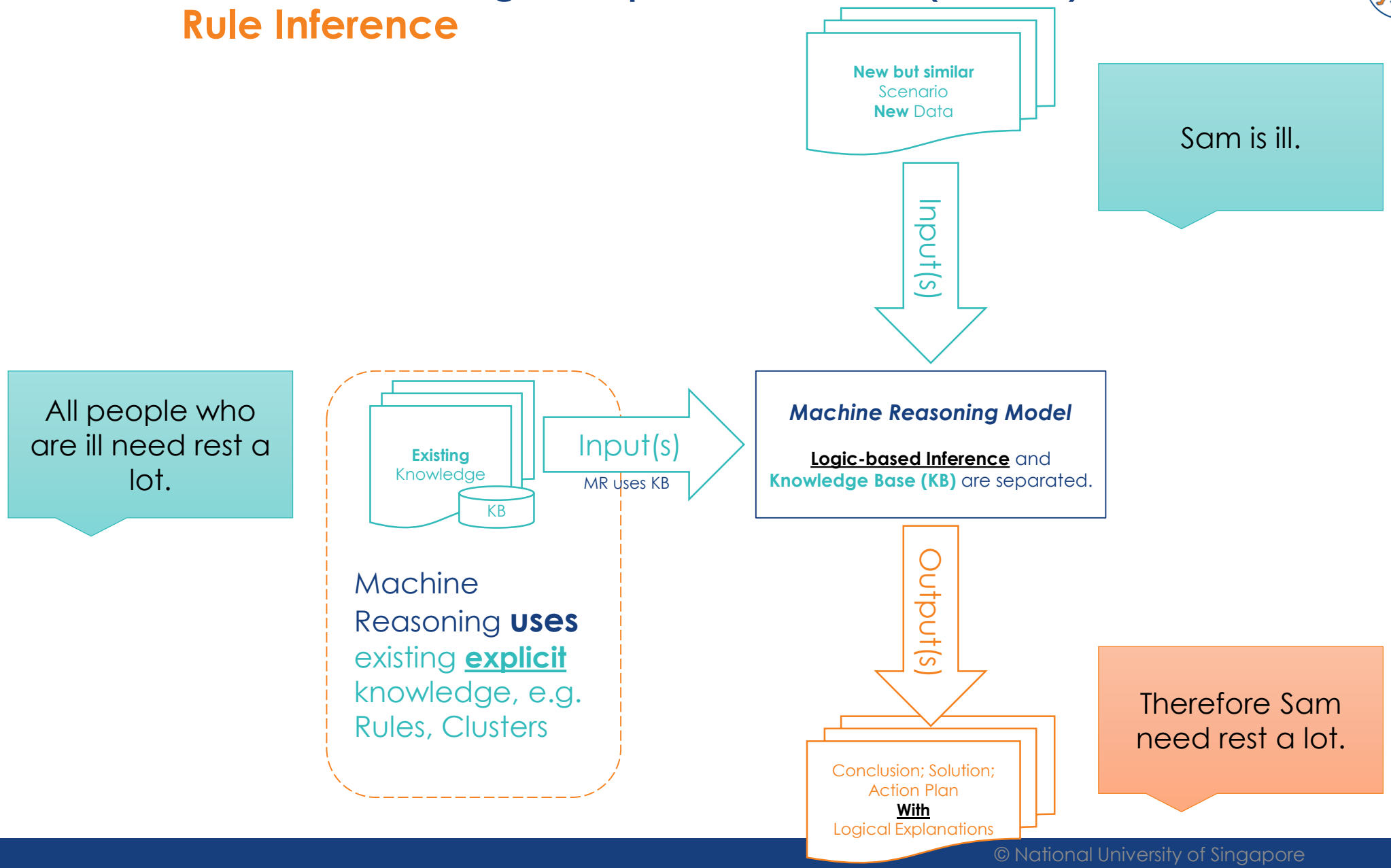
Rule Inference

- **Knowledge/Rule** : **All people who are ill** need rest a lot.
- **Individual 1** : **Sam** is **ill**, therefore he need rest a lot.
- **Individual 2** : **Jessie** is **ill**, therefore she need rest a lot.
- **Individual …**

☺ **Reasoning Rationality: Universal → Individual**

Forms of Knowledge Representation (Rules)

Rule Inference



Forms of Knowledge Representation (Rules)

Rules

- **Example rules in application**

IF ‘age of the customer’ < 18

AND ‘cash withdrawal’ > $1,000

THEN ‘signature of the parent’ is required

IF ‘taxable income’ > $16,238

THEN ‘Medicare levy’ = ‘taxable income’ * 1.5 %

Forms of Knowledge Representation (Rules)

Rules

- **Rules (business knowledge) in Rule/Process Reasoning System are designed as mutually independent**

  Each rule represents a single chunk of knowledge

  - **IF *pet_size*** = **medium THEN *pet_recommend*** = **cats** or **small dogs**

- **Rules are based on a priori knowledge or heuristics**

  Rules are derived from domain experts who uses experiential knowledge and "rules-of-thumb"

  - **IF *buyer*** = **female THEN *pet_recommend*** = **hamster**

- **Rules can incorporate uncertainties**

  Real life business situations are plagued with uncertainties that make decision-making difficult (or flexible)

  - **IF *buyer_work*** = **long_hours THEN *pet_recommend*** = **dog** (30% confidence in rule conclusion)

2.1 Machine Reasoning Enabler: Knowledge Representation

2.1.1 Forms of Knowledge Representation

2.1.2 Forms of Knowledge Representation (Rules)

2.1.3 Exercise

[Exercise] Machine Reasoning Enabler: Knowledge Representation

- **Convert the following knowledge about animals into WHEN/THEN rules:**
  1. animals with hair as their body covering are mammals
  2. animals that feed their young with milk are mammals
  3. animals with feathers as their body covering are birds
  4. animals that fly and reproduce by eggs are birds
  5. mammals that eat meat are carnivores
  6. mammals with pointed teeth, claws on their feet, and eyes that point forward are carnivores
  7. mammals that eat grass are herbivores
  8. mammals with hooves on their feet are herbivores
  9. carnivores that have a tawny colour and dark spots as their marking are cheetahs
  10. carnivores that have a tawny colour and dark stripes as their marking are tigers
  11. herbivores that have a tawny colour and dark spots as their marking and long necks are giraffes
  12. herbivores that have a black and white colour are zebras
  13. birds that walk and are black and white and have a long neck are ostriches
  14. birds that swim and are black and white are penguins
  15. birds that fly and are black and white are albatrosses

2.2 Machine Reasoning Enabler: Knowledge Acquisition

2.2.1 Knowledge Acquisition (manual elicitation vs auto discovery)

2.2.2 Knowledge Elicitation using Knowledge Models

2.2.3 Knowledge Discovery using Data Mining Models

2.2.4 Exercise

2.2 Machine Reasoning Enabler: Knowledge Acquisition

2.2.1 Knowledge Acquisition (manual elicitation vs auto discovery)

2.2.2 Knowledge Elicitation using Knowledge Models

2.2.3 Knowledge Discovery using Data Mining Models

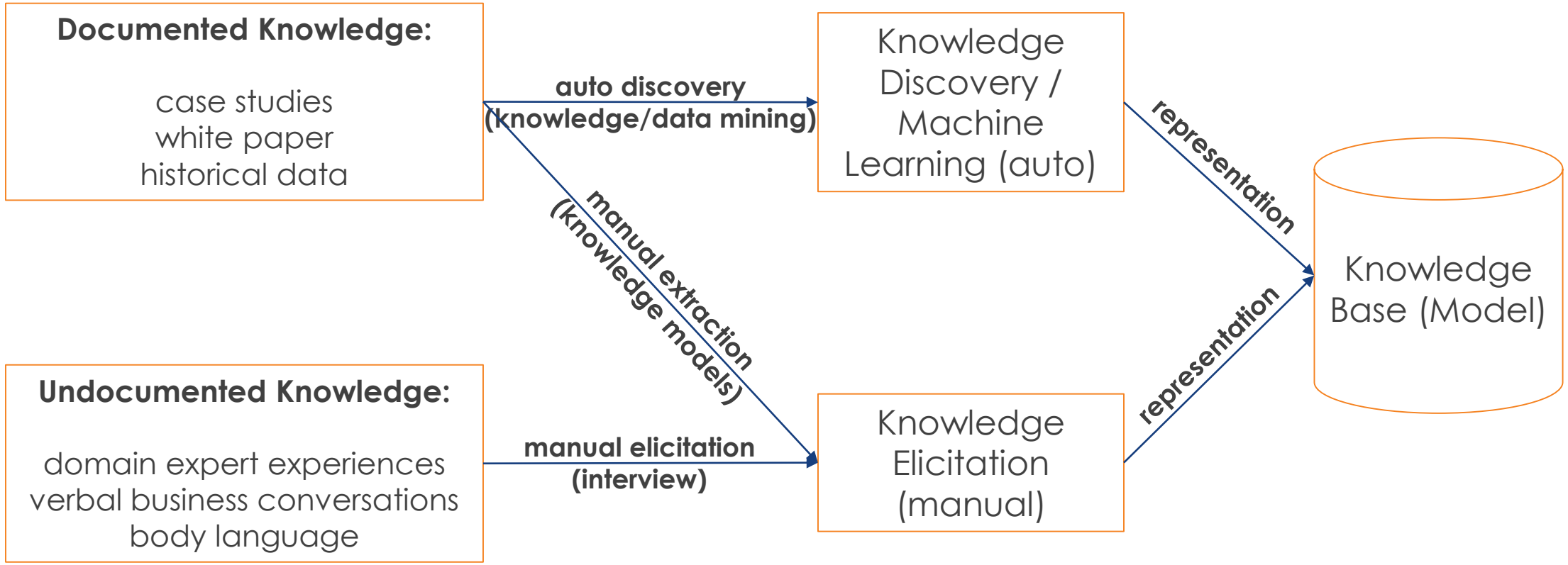
2.2.4 Exercise

Knowledge Acquisition (manual elicitation vs auto discovery)

- **Knowledge Acquisition is the transfer and transformation of problem solving knowledge into a form that can be used to build intelligent systems.**
- **Knowledge acquisition is also called:**
  - Knowledge capture
  - Knowledge elicitation
  - Requirements engineering
- **Personnel involved:**
  - Knowledge holder, e.g. subject matter expert (SME); process owner
  - Knowledge engineer, e.g. business analyst; system analyst; consultant

Knowledge Acquisition

Acquisition Methods



Knowledge Acquisition (manual elicitation)

Interview Elicitation

- **Elicitation is acquisition of tacit knowledge from a subject matter expert**
- **The Knowledge elicitation approach:**
  - Capture knowledge using interviews
  - Interpret & Analyze the transcripts and data obtained
  - Build knowledge models (knowledge representation)
  - Use the knowledge models to guide further elicitation
  - Verify & Validate the captured knowledge
  - Stop when the knowledge model is enough for building business reasoning system

Knowledge Acquisition (manual elicitation)

Interview Best Practices

- **More Beneficial interviewing expert at their workplace**
- **Make sure the meeting place is quiet and free from interruptions**
- **Before the interview:**
  - Do background research on the domain area
  - Background check on the domain expert
  - Design and phrase your questions
  - Email questions to domain expert
  - Acquire and prepare the tools for the interview
- **During the interview:**
  - Introductions & Social preliminaries
  - State purpose of interview
  - Give a brief on the roles and responsibilities
  - Be courteous; Listen closely; Avoid arguments
  - Investigate each topic in detail
  - Evaluate session outcome
- **Observe confidentiality**

2.2 Machine Reasoning Enabler: Knowledge Acquisition

2.2.1 Knowledge Acquisition (manual elicitation vs auto discovery)

2.2.2 Knowledge Elicitation using Knowledge Models

2.2.3 Knowledge Discovery using Data Mining Models

2.2.4 Exercise

Knowledge Elicitation using Knowledge Models

- **After acquiring domain knowledge from the experts and other sources, how do we present a comprehensive view of this knowledge?**
- **Knowledge Models (Templates for knowledge representation)**
  - A knowledge model is a group of **structured representations** of knowledge that allows us to better understand the domain and the processes involved in decision making.
  - Documented models provide rich descriptions of domain knowledge that is **independent** of any particular software implementation.
  - These models also serve as a basis for communication among stakeholders: experts, analysts, developers, and end users.

元素周期表(Periodic Table of (Chemical) Elements)

碱金属 alkali metals	碱土金属 alkaline-earth metals	镧系元素 lanthanide	锕系元素 actinicles	过渡金属 transition metal
主族金属 Main group metals	类金属 metalloid	非金属 nonmetal	卤素 halogen	惰性气体 inert gases

气 体 gas　液 体 liquid　固 体 solid　合成元素 composite element　未知元素 unknown element

1 氢 H Hydrogen 1.0079																	2 氦 He Helium 4.0026
3 锂 Li Lithium 6.941	4 铍 Be Beryllium 9.012											5 硼 B Boron 10.811	6 碳 C Carbon 12.011	7 氮 N Nitrogen 14.007	8 氧 O Oxygen 15.999	9 氟 F Fluorine 18.998	10 氖 Ne Neon 20.17
11 钠 Na Sodium 22.989	12 镁 Mg Magnesium 22.989											13 铝 Al Aluminum 26.982	14 硅 Si Silicon 28.805	15 磷 P Phosphorus 30.974	16 硫 S Sulfur 32.06	17 氯 Cl Chlorine 35.453	18 氩 Ar Argon 39.94
19 钾 K Potassium 39.098	20 钙 Ca Calcium 40.08	21 钪 Sc Scandium 44.956	22 钛 Ti Titanium 47.9	23 钒 V Vanadium 50.9415	24 铬 Cr Chromium 51.996	25 锰 Mn Manganese 54.938	26 铁 Fe Iron 55.84	27 钴 Co Cobalt 58.9332	28 镍 Ni Nickel 58.69	29 铜 Cu Copper 63.54	30 锌 Zn Zinc 65.38	31 镓 Ga Gallium 69.72	32 锗 Ge Germanium 72.5	33 砷 As Arsenic 74.922	34 硒 Se Selenium 78.9	35 溴 Br Bromine 79.904	36 氪 Kr Krypton 83.8
37 铷 Rb Rubidium 85.467	38 锶 Sr Strontium 87.62	39 钇 Y Yttrium 88.906	40 锆 Zr Zirconium 91.22	41 铌 Nb Niobium 92.9064	42 钼 Mo Molybdenum 95.94	43 锝 Tc Technetium 99	44 钌 Ru Ruthenium 101.07	45 铑 Rh Rhodium 102.906	46 钯 Pd Palladium 106.42	47 银 Ag Silver 107.868	48 镉 Cd Cadmium 112.41	49 铟 In Indium 114.82	50 锡 Sn Tin 118.6	51 锑 Sb Antimony 121.7	52 碲 Te Tellurium 127.6	53 碘 I Iodine 126.905	54 氙 Xe Xenon 131.3
55 铯 Cs Cesium 132.905	56 钡 Ba Barium 137.33	71 镥 Lu Lutetium 174.96	72 铪 Hf Hafnium 178.4	73 钽 Ta Tantalum 180.947	74 钨 W Tungsten 183.8	75 铼 Re Rhenium 186.207	76 锇 Os Osmium 190.2	77 铱 Ir Iridium 192.2	78 铂 Pt Platinum 195.08	79 金 Au Gold 196.967	80 汞 Hg Mercury 200.5	81 铊 Tl Thallium 204.3	82 铅 Pb Lead 207.2	83 铋 Bi Bismuth 208.98	84 钋 Po Polonium (209)	85 砹 At Astatine (201)	86 氡 Rn Radon (222)
87 钫 Fr Francium (223)	88 镭 Ra Radium 226.03	103 铹 Lr Lawrencium 260	104 𨧀 Rf Rutherfordium (261)	105 𨨏 Db Dubnium (262)	106 𨭆 Sg Seaborgium (263)	107 𨨨 Bh Bohrium (262)	108 𨭉 Hs Hassium (265)	109 鿏 Mt Meitnerium (266)	110 𫟼 Ds Darmstadtium (269)	111 𬬭 Rg Roentgenium (272)	112 Uub (277)	113 Uut 284	114 Uuq 289	115 Uup 288	116 Uuh 292	117 Uus unknow	118 Uuo 294

镧系 Lanthanide (Lanthanoid)	57 镧 La Lanthanum 138.905	58 铈 Ce Cerium 140.12	59 镨 Pr Praseodymium 140.91	60 钕 Nd Neodymium 144.2	61 钷 Pm Promethium 147	62 钐 Sm Samarium 150.4	63 铕 Eu Europium 151.96	64 钆 Gd Gadolinium 157.25	65 铽 Tb Terbium 158.93	66 镝 Dy Dysprosium 162.5	67 钬 Ho Holmium 164.93	68 铒 Er Erbium 167.2	69 铥 Tm Thulium 168.943	70 镱 Yb Ytterbium 173.0
锕系 Actinicles	89 锕 Ac Actinium 227.03	90 钍 Th Thorium 232.04	91 镤 Pa Protactinium 231.04	92 铀 U Uranium 238.03	93 镎 Np Neptunium 237.05	94 钚 Pu Plutonium 244	95 镅 Am Americium 243	96 锔 Cm Curium 247	97 锫 Bk Berkelium 247	98 锎 Cf Californium 251	99 锿 Es Einsteinium 254	100 镄 Fm Fermium 257	101 钔 Md Mendelevium 258	102 锘 No Nobelium 259

Knowledge Elicitation using Knowledge Models

Document Templates

- **Concept Dictionary**
- **Concept Tree**
- **Composition Tree**
- **Decision Tree** — KIE Guided Decision Tree
- **Rules & Decision Table** — KIE Guided Rules; Decision Table
- **Data Model** — KIE Data Model: Object; Field; Type
- **Flowchart (Workflow)** — KIE Process: Task level for Business Functions
- **Activity Flow Diagram** — KIE Process: Task level for Business Teams/Roles
- **RACI Matrix** — KIE Business Teams/Departments/Roles/Groups

Knowledge Elicitation using Knowledge Models

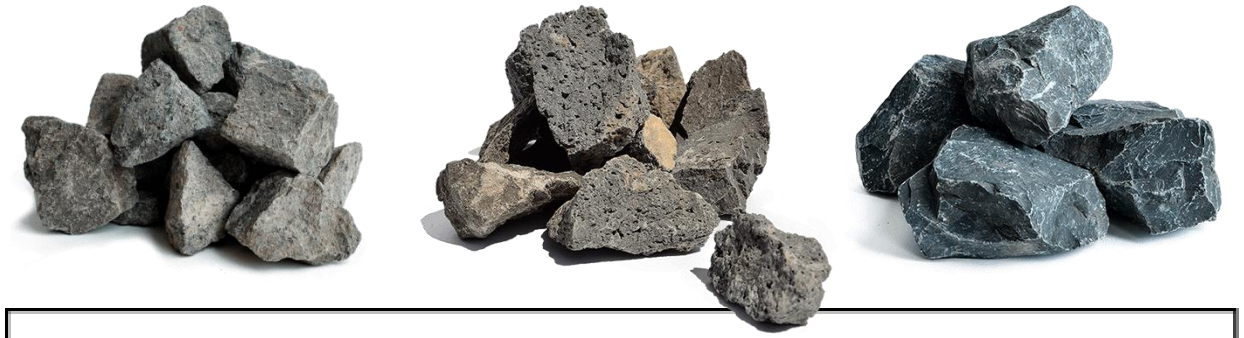
Concept Dictionary

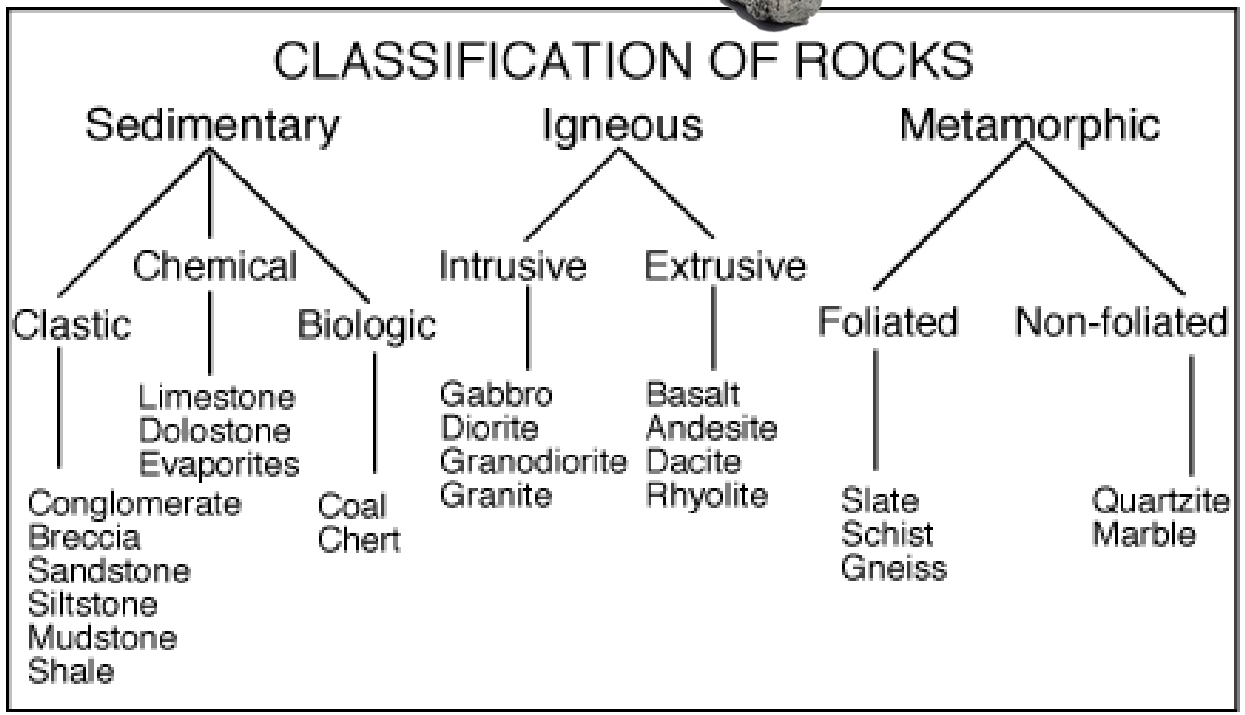
Concept	Synonyms Abbr	Meaning
Esophagus	Gullet Oesophagus	Sometimes known as the gullet. Muscular tube through which food passes from pharynx to the stomach
Duodenum		The first section of the small intestine
Peptic ulcer	PUD	Area of the gastrointestinal tract that is extremely painful. Mucosal erosions equal to or greater than 0.5cm.
Hyperacidity	Acid dyspepsia, Amalpitta	A condition of excreting more than the normal amount of hydrochloric acid in the stomach

- **A concept dictionary contains the list of all relevant concepts that are used in the problem domain to solve the problem.**
- **The dictionary provides a detailed explanation of the concept and can include any information that is useful for a good understanding of the concept.**
- **It can be similar to a glossary.**
- **It does not have any particular format.**

Knowledge Elicitation using Knowledge Models

Concept Tree

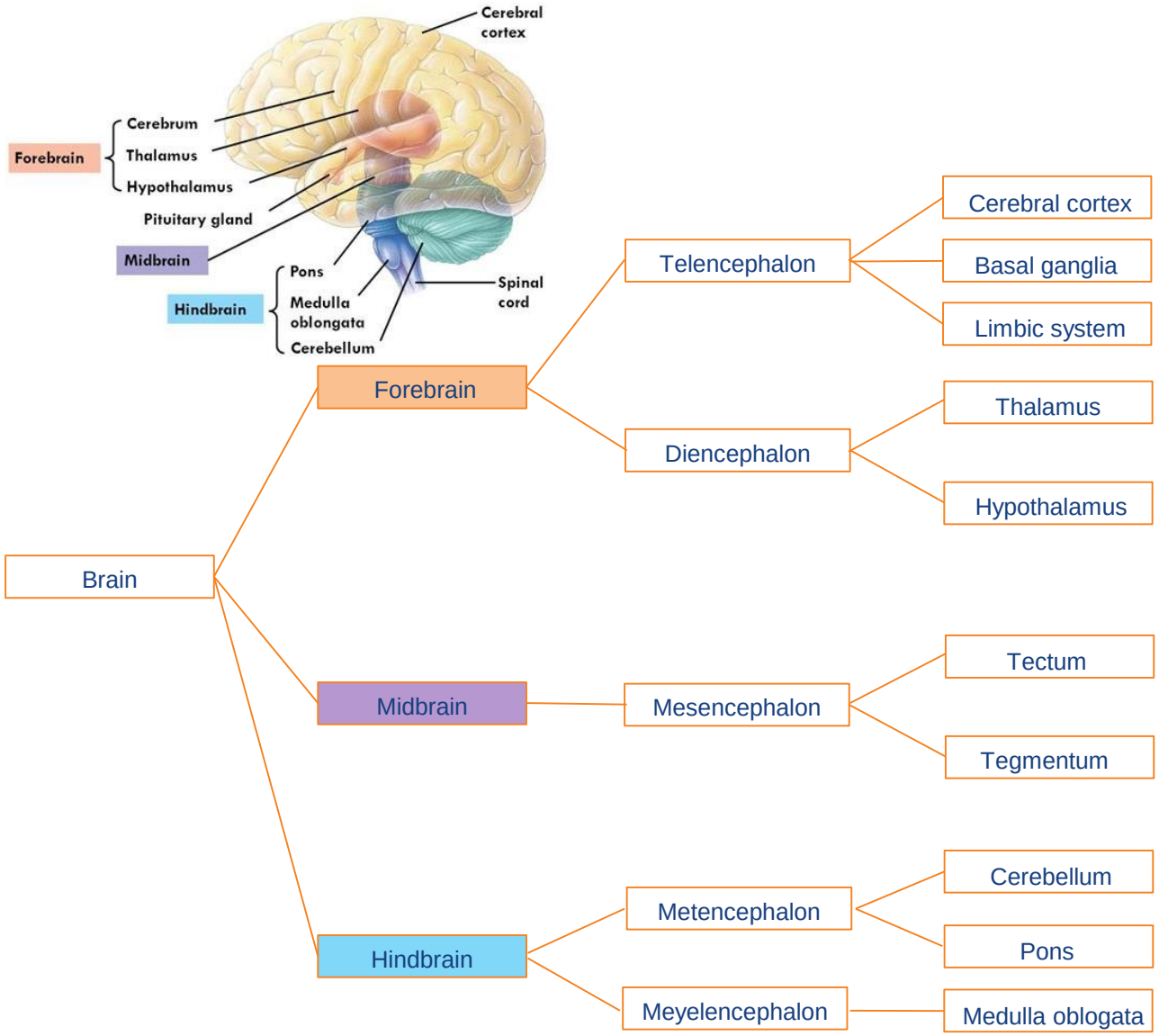




- Tree that shows concepts and the classes and sub-classes.
- All relationships must be **"is a"**.
- Check the tree by looking at the lowest and highest nodes and asking **"is <sub-concept> a (type of) <concept>"**.
- Nodes should have clear & complete names.
- Captured terminology and landscape of the domain.

Knowledge Elicitation using Knowledge Models

Composition Tree

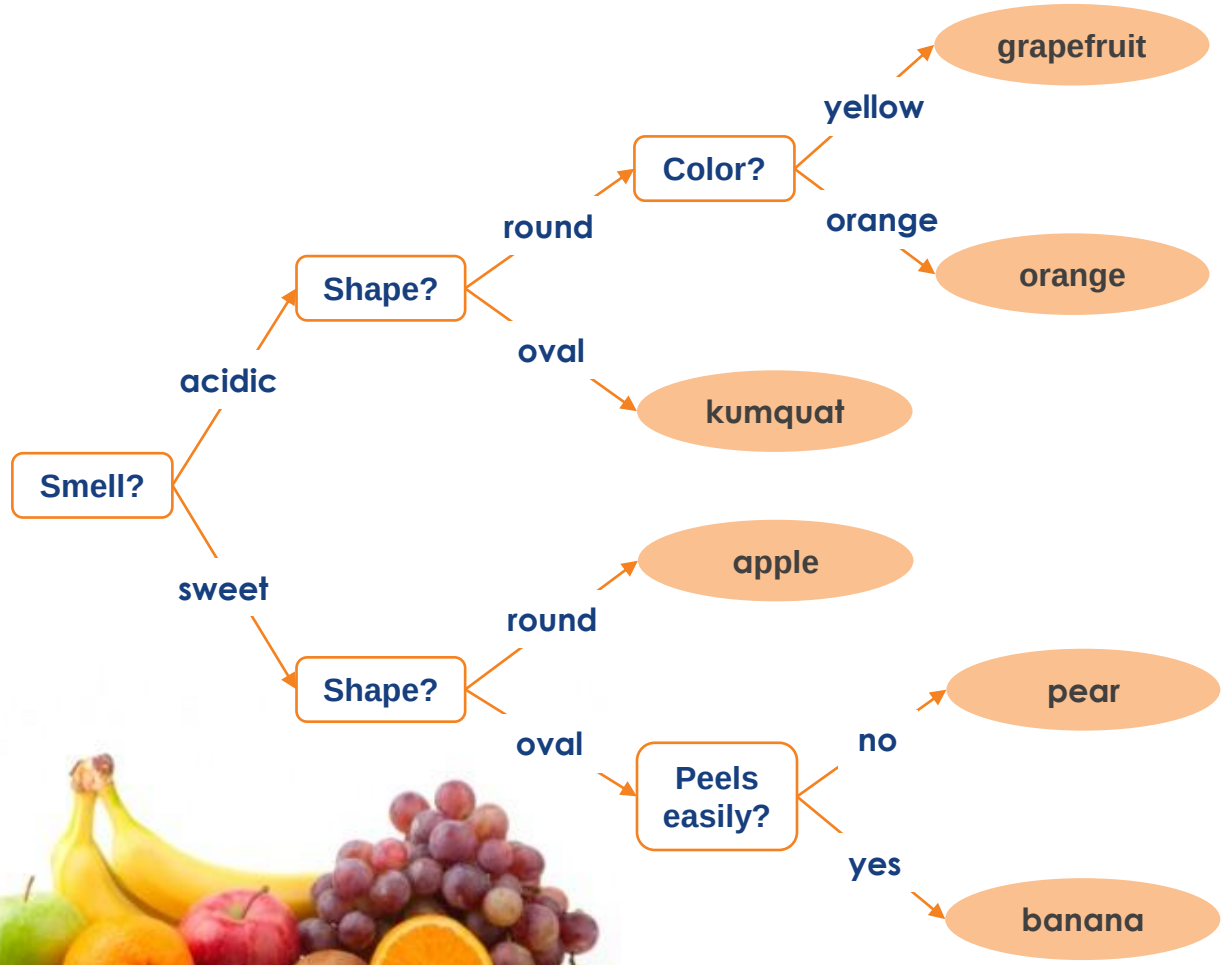


- **Detailed concept breakdowns into its constituent parts.**
- **All relationships must be "is part of".**
- **Understand things as**
  - Products (parts of a machinery)
  - Organisations (your organisation chart)
  - Documents (the table of contents)

http://blog.askiitians.com/wp-content/uploads/2016/05/Brain-1.jpg

Knowledge Elicitation using Knowledge Models

Decision Tree



KIE Guided Decision Tree

- Tree shows the alternative courses of action or casual consequences for a particular decision.
- Condition/Rule based domain knowledge
- A snapshot of the experts knowhow

Knowledge Elicitation using Knowledge Models

Rules & Decision Table KIE Guided Rules; Decision Table

Rule No.	Condition 1	Logical Operand	Condition 2	Sub-goal
F-1	franchise-fee ≤ threshold1	AND	royalty ≤ threshold2	Franchise = ok
F-2	franchise-fee ≤ threshold1	AND	royalty > 20% x franchise-fee	Franchise = not-ok
F-?	…	…	…	…

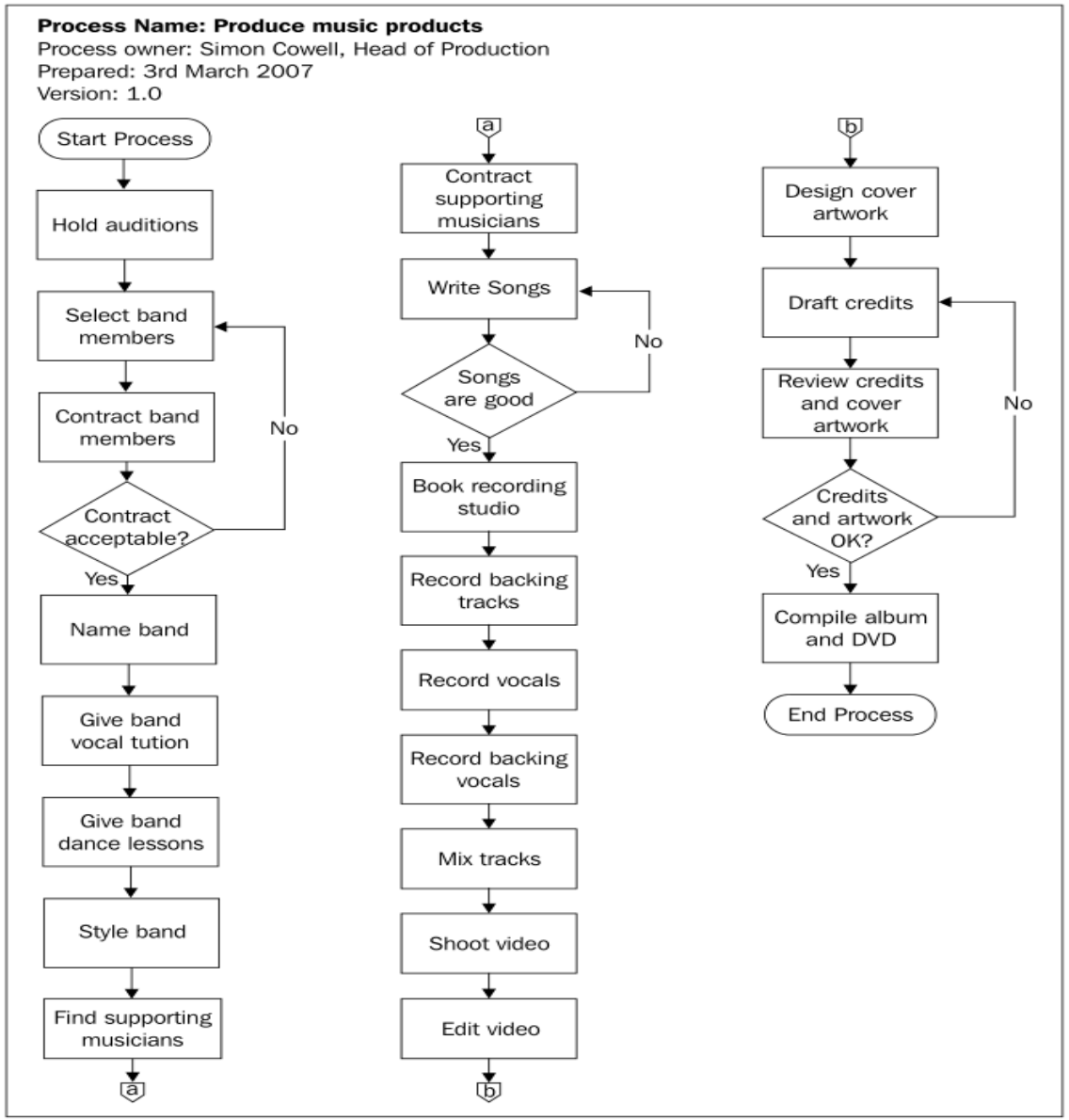
Knowledge Elicitation using Knowledge Models

Data Model — KIE Data Model: Object; Field; Type

Sub-goal	Attribute	Inferable or Observable	KIE Field Type & Value			English Translation
KIE Data Model: Data Object	**KIE Data Model: Object Field**	**KIE Form: User Interface**	**String, Integer, Float, Boolean, Date, etc.**	**Value Range**	**Value Unit**	**KIE Data Model: Comments**
Franchise	franchise-fee	Observable	Float	1 - 50,000	SGD $	The price to be paid for the franchise
	royalty	Observable	Float	1 - 10,000	SGD $	The monthly fee payable to franchisor
	lease-period	Observable	Integer	1 - 5	years	The Franchise Lease period
Profit	**Income**	**Inferable**	Float	1 – 1,000,000	SGD $	Annual income from sale of goods
	Expenditure	Inferable	Float	1 – 1,000,000	SGD $	Annual expenditure from sale of goods
Income	goods	Observable	Float	1 – 1,000,000	SGD $	The sales proceeds from goods sold
	services	Observable	Float	1 – 1,000,000	SGD $	Sales proceeds from services rendered
Location	in-city	Observable	Boolean	True or False	N.A.	Planned shop in city area
	in-suburb	Observable	Boolean	True or False	N.A.	Planned shop in suburb area

Knowledge Elicitation using Knowledge Models

Flowchart KIE Process: Task level for Business Functions

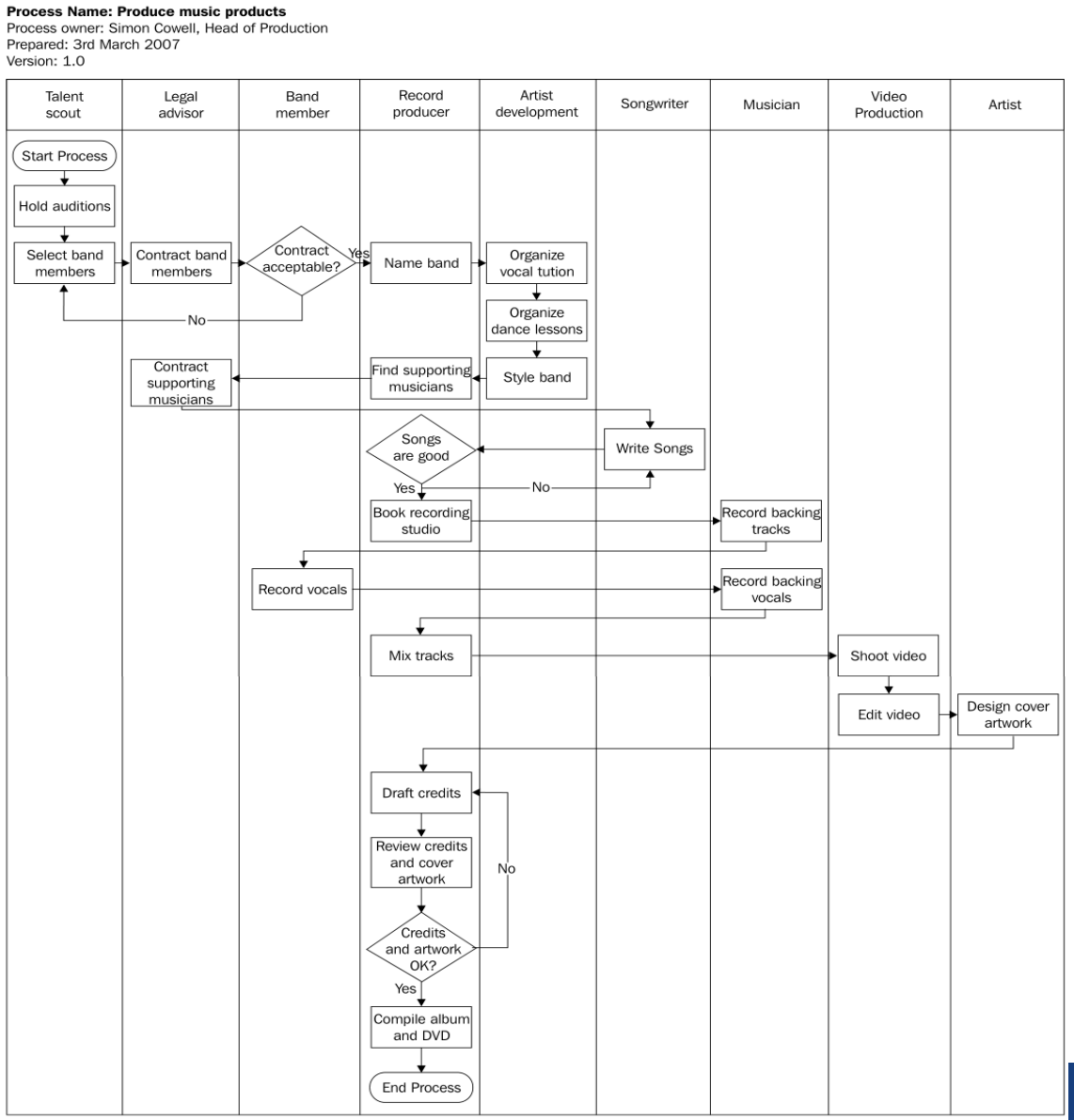


- The key to develop Flowchart to model/capture business process workflow, is to define the **sequence of activities**, and to identify those points where the flow can go two ways, depending on the circumstances.
- Write the activity name in as few words as possible, e.g. **Verb + Noun** pairs
- Write **decision points** as clear questions to which the answer is either "yes/true" or "no/false".

Knowledge Elicitation using Knowledge Models

Activity Flow Diagram

KIE Process: Task level for Business Teams/Roles



- Activity Flow Diagram captures "**who does what (activity)**" in the workflow.
- Identify roles and responsibilities (**Swimlanes**)
- A single **activity** should map to a single **role**. If this doesn't seem possible, then consider whether the activity should actually be split out into multiple activities.
- Expand flowchart by drawing swimlanes for each activity under the identified roles/teams.

Knowledge Elicitation using Knowledge Models

RACI Matrix KIE Business Teams/Departments/Roles/Groups

Process step / Role	Hold auditions	Select band members	Contract band members	Name band	Organise vocal tuition	Organise dance lessons	Stylise band	Find supporting musicians	Contract supporting musicians	Write songs	Book recording studio	Record backing vocals	Record vocals	Record backing vocals	Mix tracks	Shoot video	Edit video	Design cover artwork	Draft credits	Review credits and cover artwork	Compile album and DVD
Note	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Talent Scout	AR	AR	R					C													
Legal Advisor			AR						AR											R	
Band Member				I	I	I	I				I		R			R				R	
Record Producer			C	AR			C	AR	R	C	AR	AR	AR	AR	AR	R	C	R	AR	R	AR
Artist Development				C	AR	AR	AR			C								C		R	
Songwriter										AR										R	
Musician											I	R		R						R	
Video Production																AR	AR			R	R
Artist																		AR		R	

Key	
R - Responsible	Actually completes the activity - responsibility can be shared. Degree of responsibility is determined by the "A".
A - Accountable	Has Yes/No authority - there can only be one "A" per activity
C - Consulted	Involved prior to decision or action - two-way communication
I - Informed	Needs to know of the decision or action - one-way communication.

- **Responsible; Accountable; Consulted; Informed**
  - **R** for responsible means, "the person who actually does the activity". Responsibility for an activity can be shared, if necessary.
  - **A** for accountable means, "the buck stops here", and the role has ultimate yes/no authority.
  - **C** for consulted means, "kept in the loop", and implies two-way communication prior to the activity.
  - **I** for informed means, "kept in the picture", and implies one-way communication after the activity.
- **Best Practices:**
  - There can only be one accountability (**A**) per activity.
  - Recommend one responsibility (**R**) only per activity.
  - Roles can combine both accountability and responsibility for activities.
  - Minimize the number of consults (**C**) and informs (**I**).
  - Authority must accompany accountability.
  - Don't map decision points on the RACI matrix, only activities.

2.2 Machine Reasoning Enabler: Knowledge Acquisition

2.2.1 Knowledge Acquisition (manual elicitation vs auto discovery)

2.2.2 Knowledge Elicitation using Knowledge Models

2.2.3 Knowledge Discovery using Data Mining Models

2.2.4 Exercise

Knowledge Discovery using Data Mining

Build intuitions

- Deductive Reasoning (Decision Rule/Tree Model)
- Inductive Reasoning (Topic Summarization/Pattern Recognition)
- Analogical Reasoning (Similarity) (FAQ Bot: Approximate String Matching of Queries)

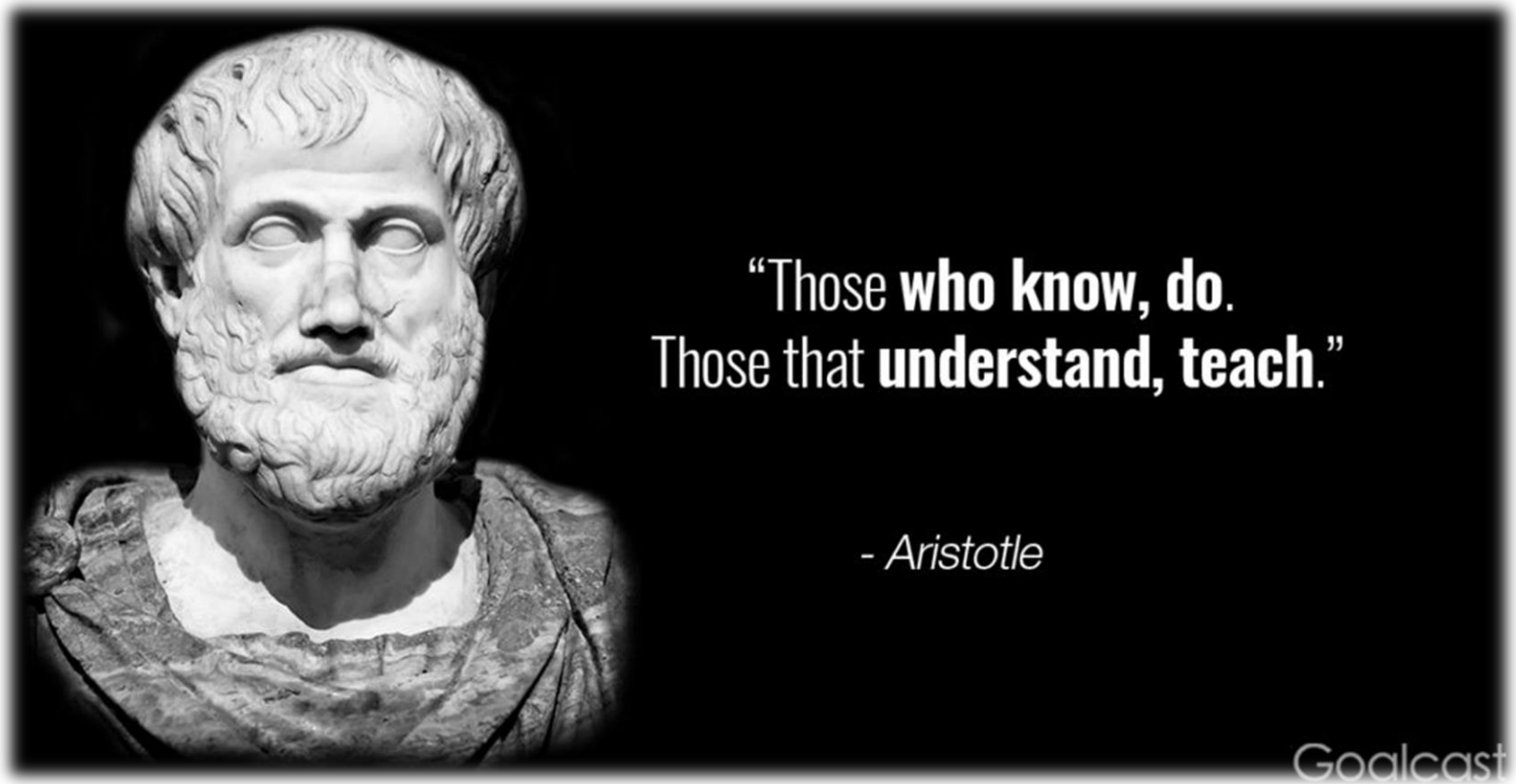
Reasoning under Uncertainty

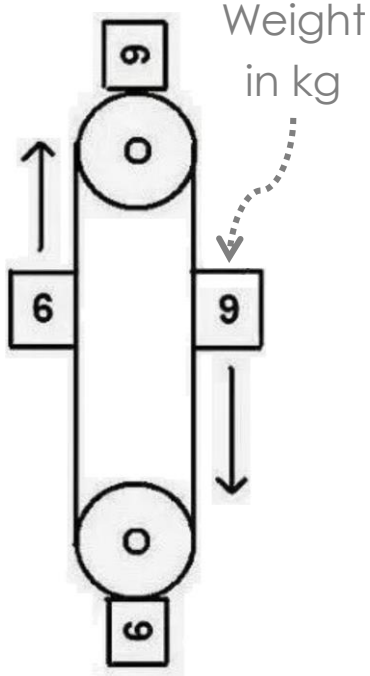
- Analogical Reasoning (Similarity) (k Nearest Neighbors)
- Approximate Reasoning (Fuzzy Logic)
- Abductive Reasoning (Probability) (Bayes Model)

Common Forms of Reasoning

1. Deductive Reasoning

- Aristotle's syllogism; Formal logic; Knowledge Graph; If-Then business rules; Declarative programing language like SQL; (Universal → Individuals)





Sam's perpetual motion machine on sale! $0.99 only!

Knowledge Discovery using Data Mining

Deductive Reasoning

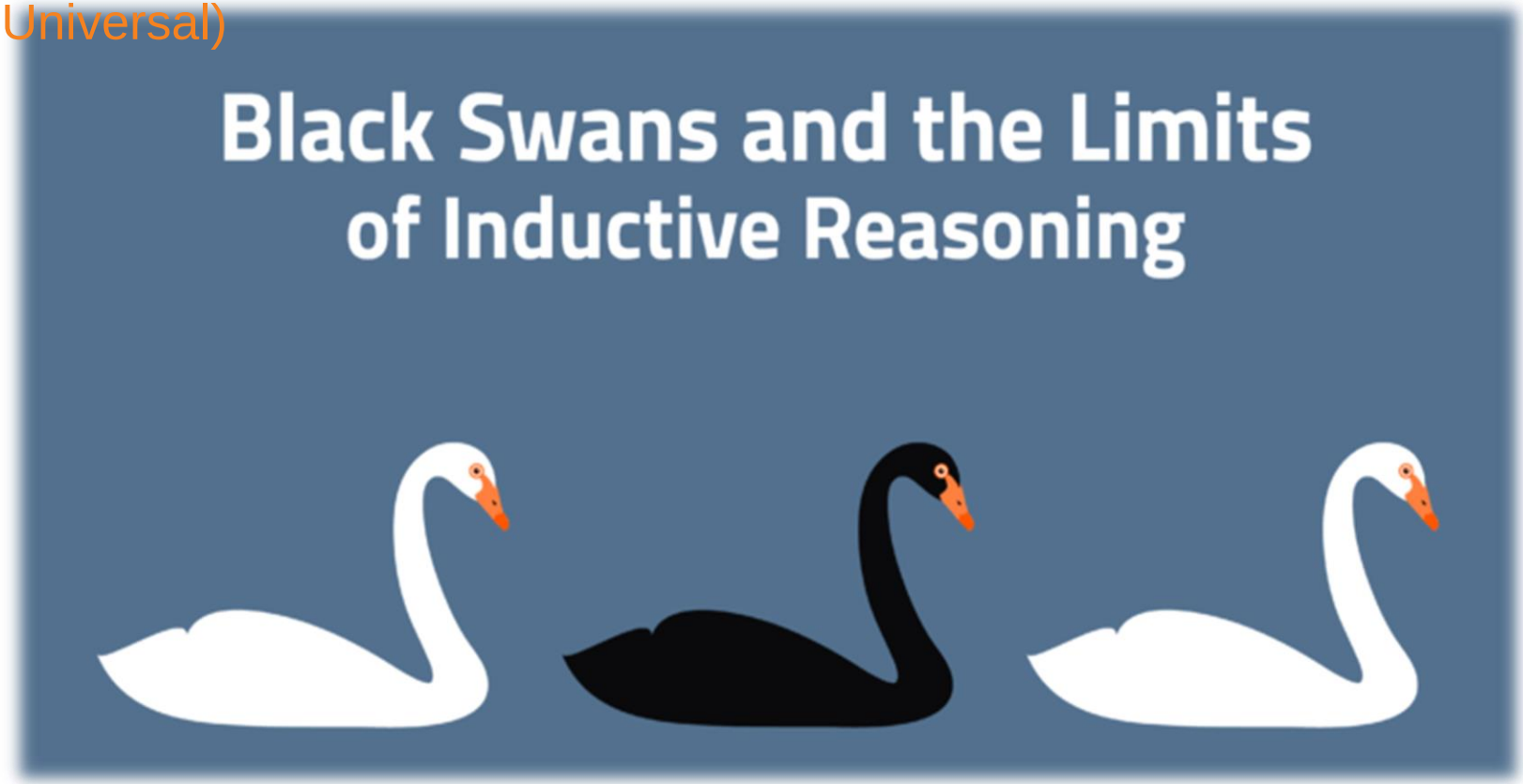
- **Knowledge/Rule** : **All people** who are **ill**, they rest a lot.
- **Individual 1** : **Sam** is **ill**, therefore he rest a lot.
- **Individual 2** : **Jessie** is **ill**, therefore she rest a lot.
- **Individual …**

☺ **Reasoning Rationality: Universal → Individual**

Common Forms of Reasoning

2. **Inductive Reasoning** (aka. learning)

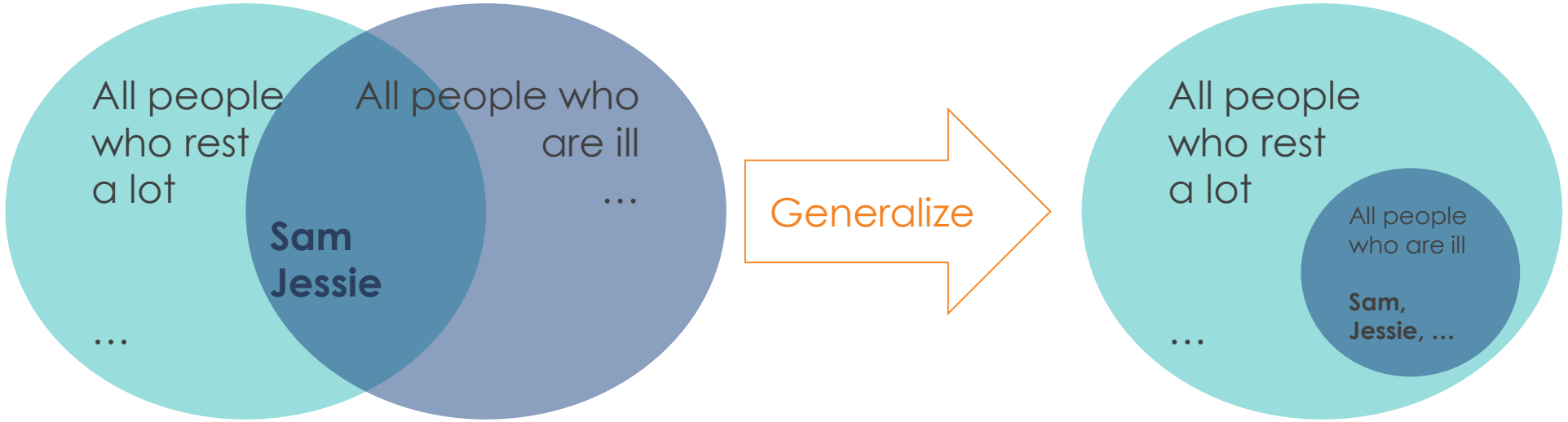
- Use meta knowledge to generate new knowledge using statistical method: learning / pattern recognition algorithms; central limit theorem; regression; (Individuals → Universal)



Knowledge Discovery using Data Mining

Inductive Reasoning

- **Individual 1** : When **Sam** is **ill**, he rests a lot.
- **Individual 2** : When **Jessie** is **ill**, she rests a lot.
- **Generalised Rule** : **All people** who are **ill**, they rest a lot.

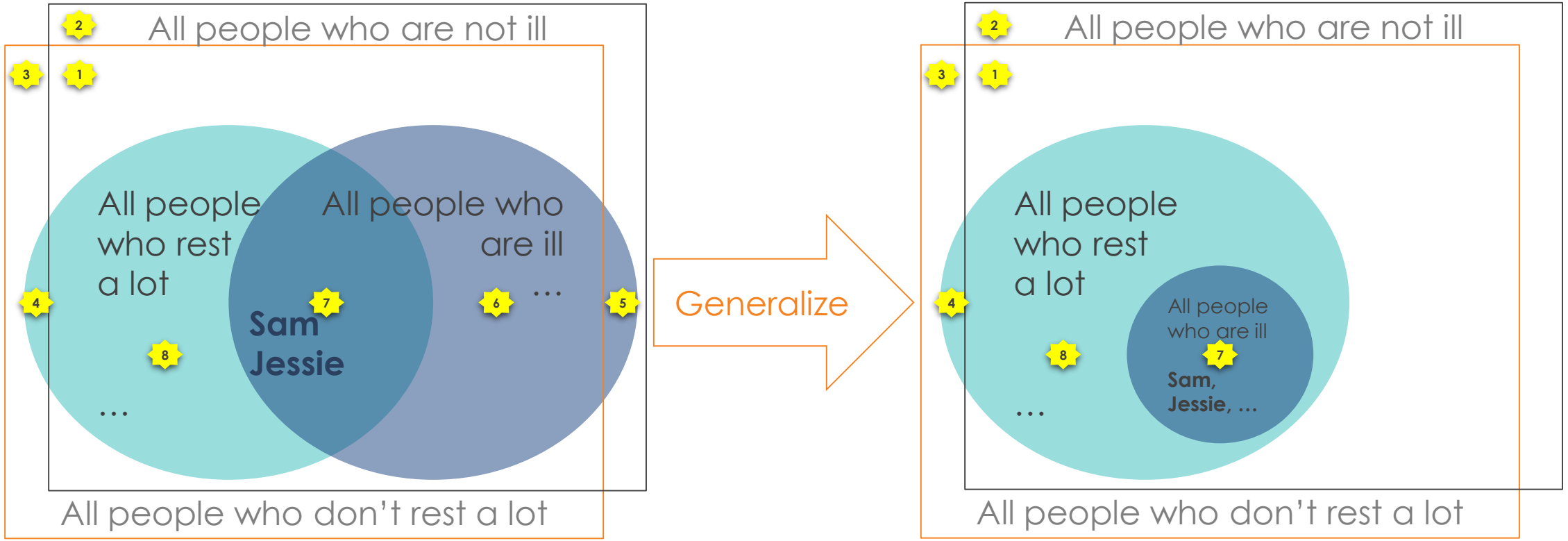


☺ **Reasoning Rationality: Individual → Universal (Machine Learning)**

Knowledge Discovery using Data Mining

Inductive Reasoning

- **Individual 1** : When **Sam** is **ill**, he rests a lot.
- **Individual 2** : When **Jessie** is **ill**, she rests a lot.
- **Generalised Rule** : **All people** who are **ill**, they rest a lot.

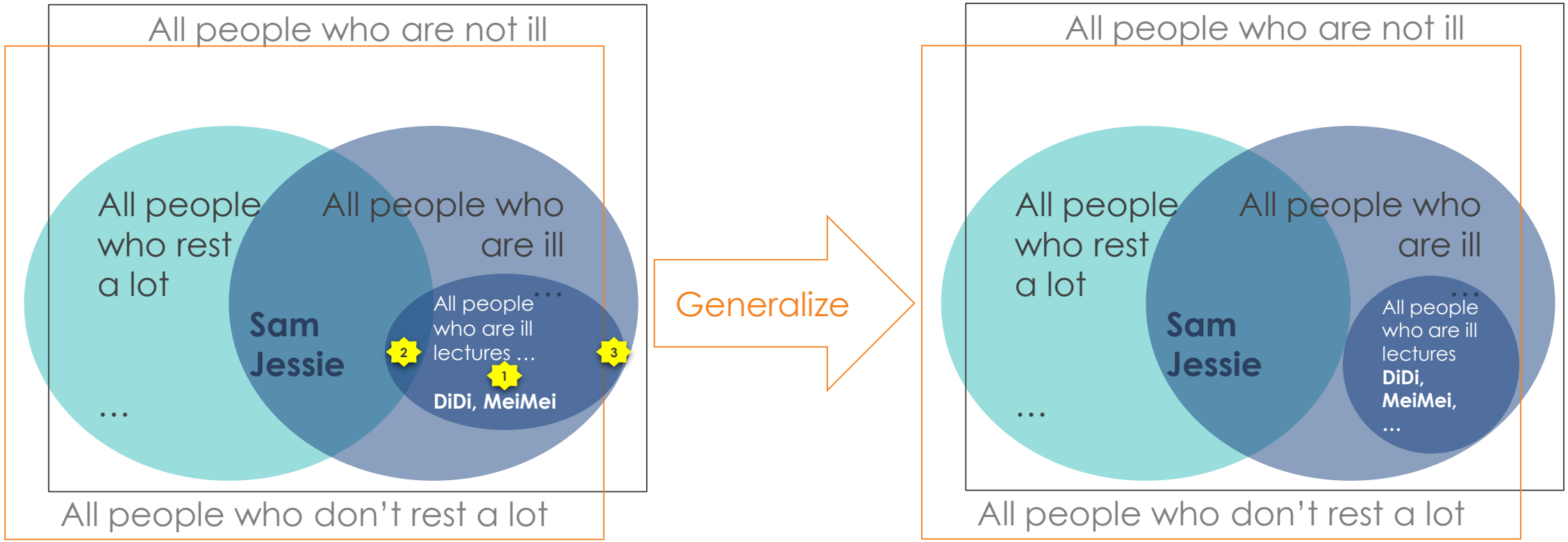


☺ **Reasoning Rationality: Individual → Universal (Machine Learning)**

Knowledge Discovery using Data Mining

Inductive Reasoning

- **Individual 1** : When **DiDi** is **ill** AND he is **lecturer**, he doesn't rest a lot.
- **Individual 2** : When **MeiMei** is **ill** AND she is **lecturer**, she doesn't rest a lot.
- **Generalised Rule** : **All people** who are **ill lecturers**, they don't rest a lot.

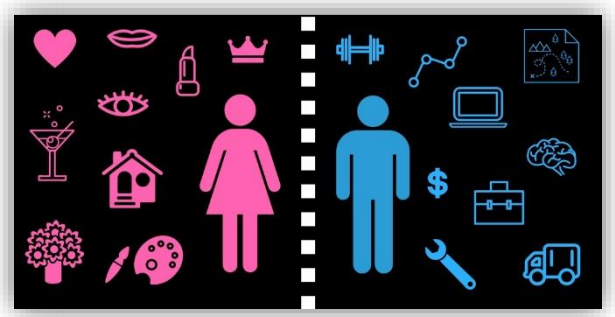


☺ **Reasoning Rationality: Individual → Universal (Machine Learning)**

Common Forms of Reasoning

3. Analogical Reasoning

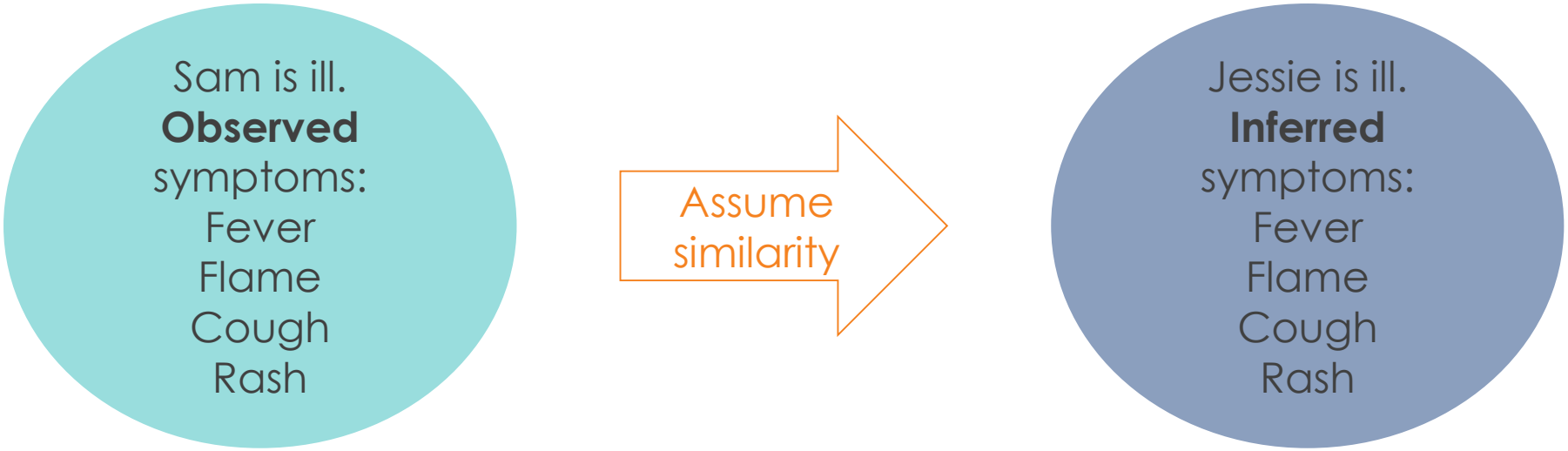
- Similarity based reasoning; Case based; K nearest neighbour; (including customer profiling for recommendation; even stereotyping)



Knowledge Discovery using Data Mining

Analogical Reasoning

- **Known case** : **Sam** is ill with his symptoms (features): fever, flame, cough, and rash.
- **Inferred case** : **Jessie** is ill too, therefore she **would** have **same** symptoms as **Sam**: fever, flame, cough, and rash.



☺ **Reasoning Rationality: Known case → Inferred case**

Common Forms of Reasoning

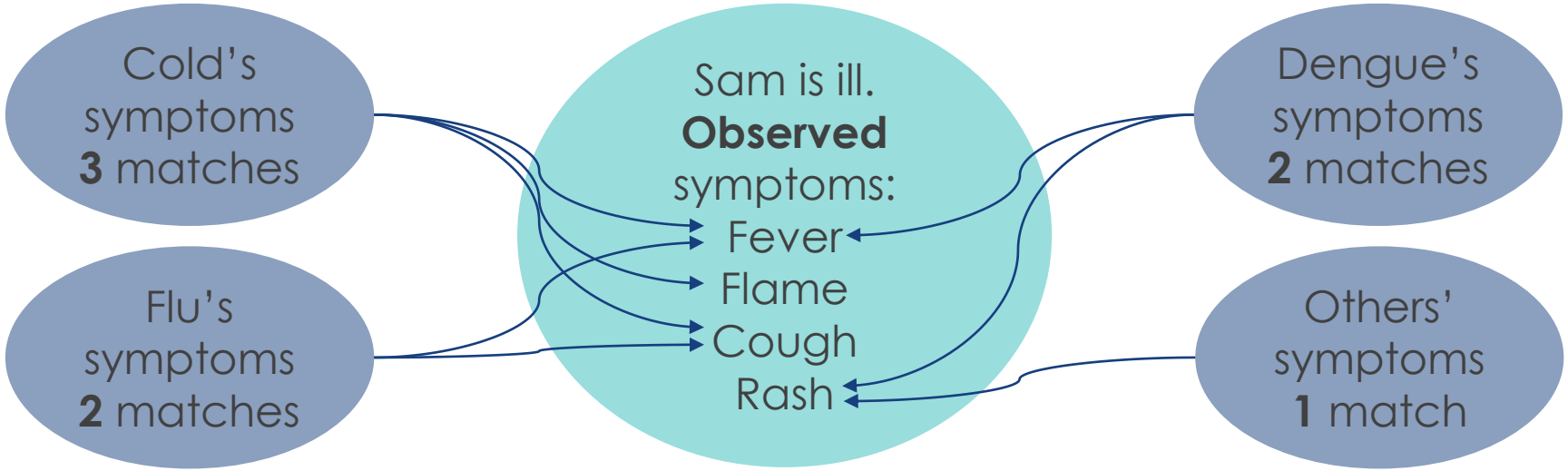
4. Abductive Reasoning

- Probabilistic calculation; Prior/Conditional/Joint probability; Bayesian network; (Hypothesis ~ Evidence)

Knowledge Discovery using Data Mining

Abductive Reasoning

- **Known observations** : **Sam** is ill with his symptoms (evidences): fever, flame, cough, and rash.
- **Inferred root cause** (Hypothesis) : **Cold? Flu? Dengue? Others?**



☺ **Reasoning Rationality: Observations → Causes likelihood**

Knowledge Discovery using Data Mining

Approximate Reasoning (Fuzzy Logic)

Long Hair Group ← **Hair length ≥ 10 cm**

Hair length < 10 cm → Short Hair Group

Long Hair Group ← **Hair length is long**

Hair length is short → Short Hair Group

What if the hair length is both long and short → Which Group?

Knowledge Discovery using Data Mining

Example model/algorithm

- **Deductive Reasoning (Decision Rule/Tree Model)**
- **Inductive Reasoning (Topic Summarization/Pattern Recognition)**
- **Analogical Reasoning (Similarity) (FAQ Bot: Approximate String Matching of Queries)**

Reasoning under Uncertainty

- **Analogical Reasoning (Similarity) (k Nearest Neighbors)**
- **Approximate Reasoning (Fuzzy Logic)**
- **Abductive Reasoning (Probability) (Bayes Model)**

Knowledge Discovery using Data Mining

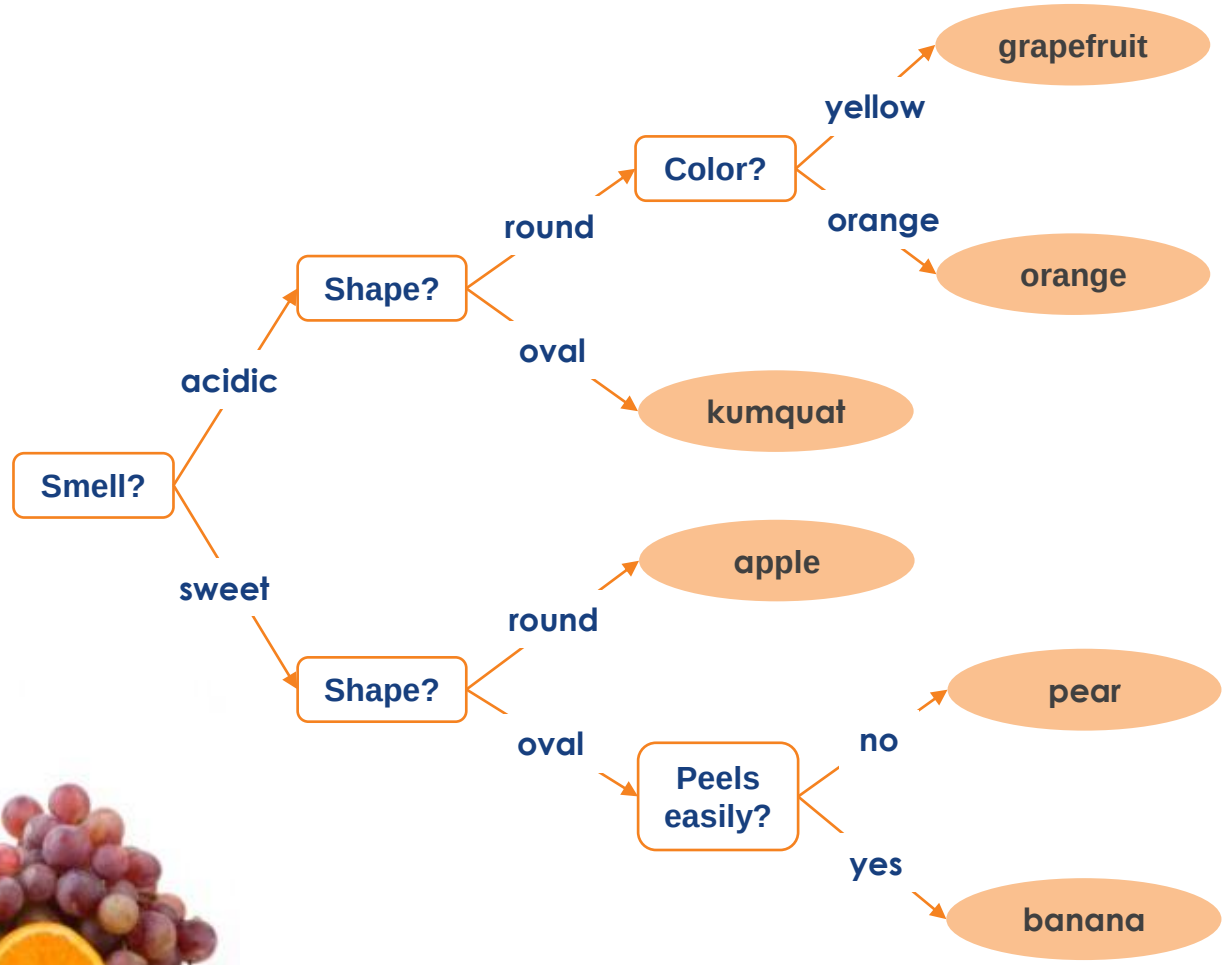
Example model/algorithm

- **Deductive Reasoning (Decision Rule/Tree Model)**
- Inductive Reasoning (Topic Summarization/Pattern Recognition)
- Analogical Reasoning (Similarity) (FAQ Bot: Approximate String Matching of Queries)

Reasoning under Uncertainty

- Analogical Reasoning (Similarity) (k Nearest Neighbors)
- Approximate Reasoning (Fuzzy Logic)
- Abductive Reasoning (Probability) (Bayes Model)

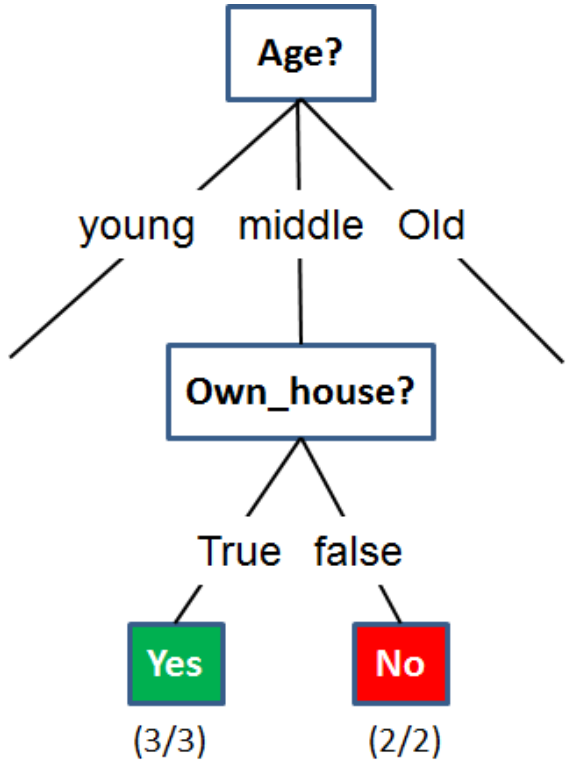
Decision Tree for fruit classification



What is Decision Tree?

- **Decision Tree (DT)** is a knowledge discovery method (supervised learning) used for classification and regression. The goal is to create a model (knowledge/rules) that predicts the value of a target variable by learning decision tree/rules inferred from the data features, through statistical calculation.

ID	Age	Has_job	Own_house	Credit_rating	Outcome
1	young	False	False	fair	No
2	young	False	False	good	No
3	young	True	False	good	Yes
4	young	True	True	fair	Yes
5	young	False	False	fair	No
6	middle	False	False	fair	No
7	middle	False	False	good	No
8	middle	True	True	good	Yes
9	middle	False	True	excellent	Yes
10	middle	False	True	excellent	Yes
11	old	False	True	excellent	Yes
12	old	False	True	good	Yes
13	old	True	False	good	Yes
14	old	True	False	excellent	Yes
15	old	False	False	fair	No



Knowledge Discovery using Data Mining

Example model/algorithm

- Deductive Reasoning (Decision Rule/Tree Model)
- **Inductive Reasoning (Topic Summarization/Pattern Recognition)**
- Analogical Reasoning (Similarity) (FAQ Bot: Approximate String Matching of Queries)

Reasoning under Uncertainty

- Analogical Reasoning (Similarity) (k Nearest Neighbors)
- Approximate Reasoning (Fuzzy Logic)
- Abductive Reasoning (Probability) (Bayes Model)

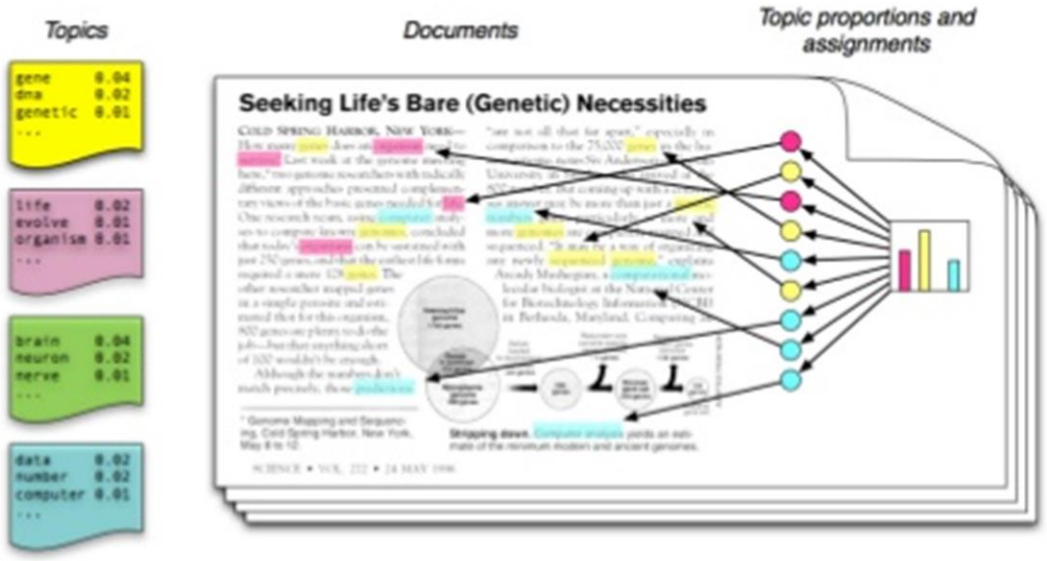
What is Topic Modeling?

- **Topic Modeling** in machine learning and natural language processing, a topic model is a type of statistical model for discovering the abstract "topics" that occur in a collection of documents. Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body. Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently: "dog" and "bone" will appear more often in documents about dogs, "cat" and "meow" will appear in documents about cats, and "the" and "is" will appear approximately equally in both.

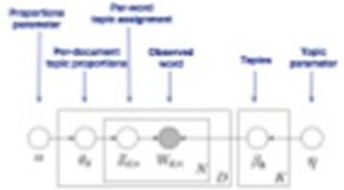
- https://en.wikipedia.org/wiki/Topic_model

Topic Modeling for Text Summarization

Latent Dirichlet Allocation



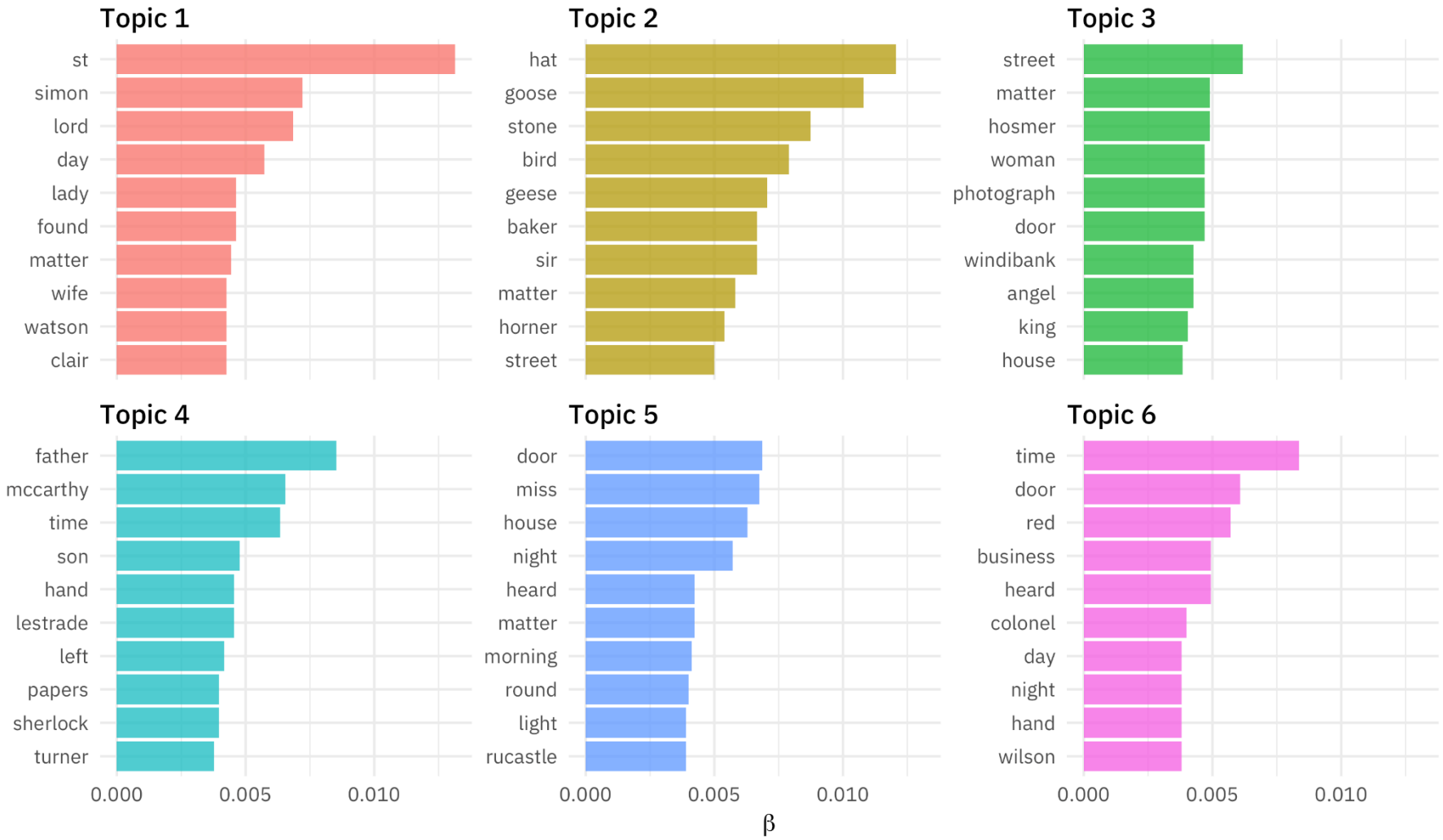
- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics



LDA is an example of a topic model for text summarization. It allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the topics.

Source https://image.slidesharecdn.com/topicmodel-12111209161l-phpapp02/95/an-introduce-to-topic-model-4-638.jpg?cb=1352711844

Topic Modeling for Text Summarization



https://juliasilge.com/blog/2018/2018-01-25-sherlock-holmes-stm_files/figure-html/unnamed-chunk-6-1.png

Knowledge Discovery using Data Mining

Example model/algorithm

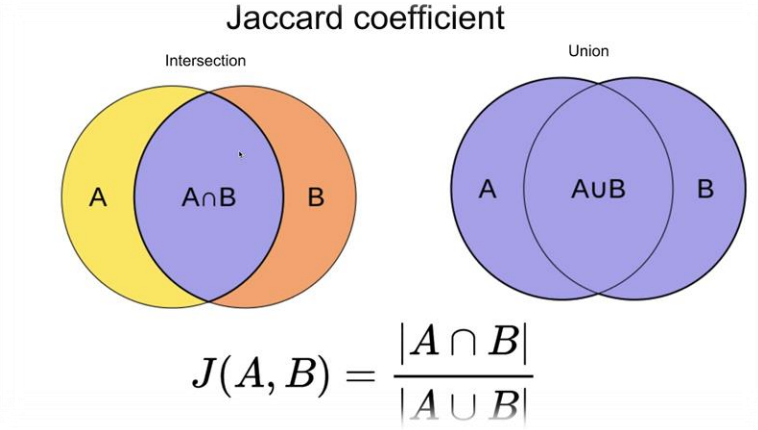
- Deductive Reasoning (Decision Rule/Tree Model)
- Inductive Reasoning (Topic Summarization/Pattern Recognition)
- **Analogical Reasoning (Similarity) (FAQ Bot: Approximate String Matching of Queries)**

Reasoning under Uncertainty

- Analogical Reasoning (Similarity) (k Nearest Neighbors)
- Approximate Reasoning (Fuzzy Logic)
- Abductive Reasoning (Probability) (Bayes Model)

What is Approximate String Matching?

- **Approximate String Matching**: In computer science, approximate string matching (often colloquially referred to as fuzzy string searching) is the technique of finding strings that match a pattern approximately (rather than exactly). A variation: similarity between two given (text) strings.
- https://en.wikipedia.org/wiki/Approximate_string_matching



Sentence A: This is machine reasoning course

Sentence B: This is reasoning systems course

A ∩ B = 4 (unique tokens)

This	is	machine	reasoning	course
This	is	reasoning	systems	course

A ∪ B = 6 (unique tokens)

This	is	machine	reasonoing	course
This	is	reasoning	systems	course

J(A, B) = 66.67% similarity

2.2 Machine Reasoning Enabler: Knowledge Acquisition

2.2.1 Knowledge Acquisition (manual elicitation vs auto discovery)

2.2.2 Knowledge Elicitation using Knowledge Models

2.2.3 Knowledge Discovery using Data Mining Models

2.2.4 Exercise

[Exercise] Machine Reasoning Enabler: Knowledge Acquisition

Exercise

- Refer to **Airport Gate Assignment System (AGAS)** case in appendix of Day 1 optional guide: **S-MR Workshop Guide.pdf**.

- Create relevant **Knowledge Models** to structurally represent/document the expert knowledge.

2.3 Knowledge Representation and Acquisition/Discovery [Workshop]

2.3.1 Deductive Reasoning (Decision Tree)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.2 Inductive Reasoning (Topic Summarization)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.3 Analogical Reasoning (FAQ Knowledge Bot)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.4 Workshop Submission

2.3 Knowledge Representation and Acquisition/Discovery [Workshop]

2.3.1 Deductive Reasoning (Decision Tree)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.2 Inductive Reasoning (Topic Summarization)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.3 Analogical Reasoning (FAQ Knowledge Bot)

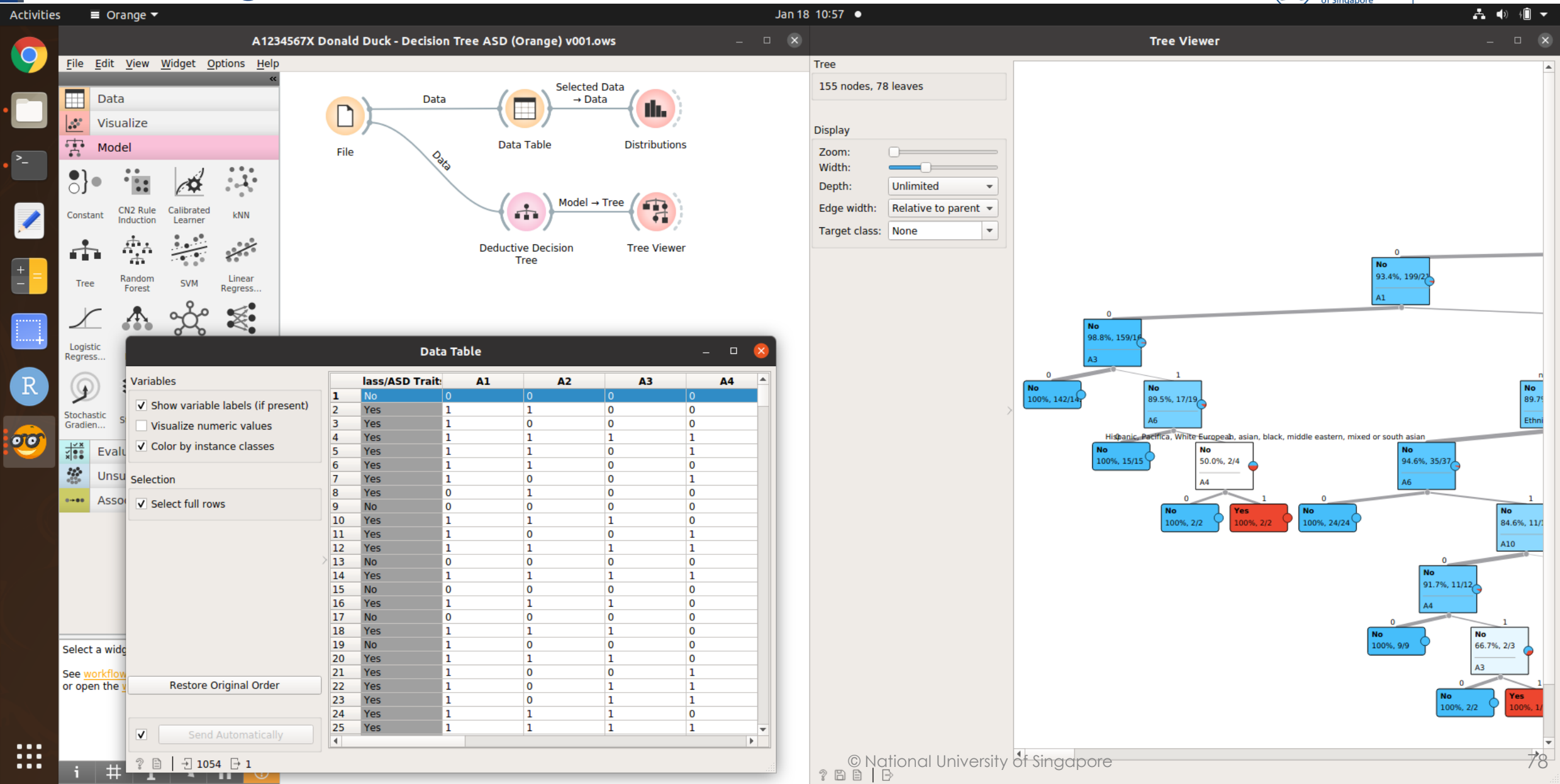
Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.4 Workshop Submission

Deductive Reasoning (Decision Tree)

- **Learning Objectives**
  - Understand how the decision tree works
  - Learn how to apply decision tree into classification and regression problem
  - Modify the model parameters and try to improve the tree
- **Package to use,**
  - Orange visual data mining tool
  - Scikit-learn 0.23.1 in Python
- **Technique,**
  - Decision tree
- **Case/Scenario,**
  - Autistic Spectrum Disorder classification
- **Requirement,**
  - Familiar with Python

Orange's Decision Tree



	lass/ASD Trait	A1	A2	A3	A4
1	No	0	0	0	0
2	Yes	1	1	0	0
3	Yes	1	0	0	0
4	Yes	1	1	1	1
5	Yes	1	1	0	1
6	Yes	1	1	0	0
7	Yes	1	0	0	1
8	Yes	0	1	0	0
9	No	0	0	0	0
10	Yes	1	1	1	0
11	Yes	1	0	0	1
12	Yes	1	1	1	1
13	No	0	0	0	0
14	Yes	1	1	1	1
15	No	0	0	0	0
16	Yes	1	1	1	0
17	No	0	0	0	0
18	Yes	1	1	1	0
19	No	1	0	0	0
20	Yes	1	1	1	0
21	Yes	1	0	0	1
22	Yes	1	0	1	1
23	Yes	1	0	1	1
24	Yes	1	1	1	0
25	Yes	1	1	1	1

Scikit's Decision Tree

- **Installation & Tutorial:**

1. Type in pip install scikit-learn in your anaconda environment
2. Read documentations about scikit-learn in https://scikit-learn.org/stable/
3. Read documentations about DT(for classification) in https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier and DT(for regression) in https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor

Workshop

- **Open the A1234567X Donald Duck – Decision Tree.ipynb python notebook file and complete the workshop**
- **Use grid search to fine tune the decision tree hyperparameters to achieve best performance in test dataset.**
- **Extract all the rules from the tree at the end of workshop.**

Reference: https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html

Workshop Submission

- **Naming convention: StudentID YourFullName.sql, e.g. A1234567X Donald Duck – sln – Decision Tree.ipynb/zip**
- **Use zip to a single file, then rename, if you plan to submit multiple files.**

References

1. User guide of Orange tool: https://orangedatamining.com/getting-started/

2. User guide of Scikit-Learn: https://scikit-learn.org/stable/user_guide.html

3. Decision Tree in Wikipedia: https://en.wikipedia.org/wiki/Decision_tree

2.3 Knowledge Representation and Acquisition/Discovery [Workshop]

2.3.1 Deductive Reasoning (Decision Tree)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.2 Inductive Reasoning (Topic Summarization)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.3 Analogical Reasoning (FAQ Knowledge Bot)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.4 Workshop Submission

Inductive Reasoning (Topic Summarization)

- **Learning Objectives**
  - Understand what is topic modeling and how the LDA (Latent Dirichlet Allocation) works
  - Learn how to apply LDA into topic modeling
- **Package to use,**
  - Sckit-learn 0.23.1 in Python
- **Technique,**
  - Latent Dirichlet Allocation
- **Case/Scenario,**
  - Autistic Spectrum Disorder FAQ
- **Requirement,**
  - Familiar with Python

Before the workshop

Installation & Tutorial:

1. **Type in pip install scikit-learn in your anaconda environment**
2. **Read documentations about scikit-learn in https://scikit-learn.org/stable/**
3. **Read documentations about LDA in https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html?highlight=latent%20dirichlet%20allocation**

Preprocess the data

Before applying LDA into your text, you should

1. **Preprocess the text (use code in day 1 'TextPreprocessing')**

2. **Vectorize each document and count words/tokens in your document with Sklearn.feature_extraction.text.CountVectorizer.**

3. **Specify how many topics/clusters)you want to obtain from the collection of documents (corpus).**

Workshop

- **Open the A1234567X Donald Duck – Topic Modeling python notebook file and complete the workshop**

- **Display and review the top 10 words in each topic, then interpret the meaning/cluster for each topic. Write down your explanation in the python notebook file.**

Workshop Submission

- **Naming convention: StudentID YourFullName.sql, e.g. A1234567X Donald Duck – sln – Topic Modeling.ipynb/zip**
- **Use zip to a single file, then rename, if you plan to submit multiple files.**

References

1. LDA in Wikipedia: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

2. User guide of LDA in scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html?highlight=latent%20dirichlet%20allocation

2.3 Knowledge Representation and Acquisition/Discovery [Workshop]

2.3.1 Deductive Reasoning (Decision Tree)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.2 Inductive Reasoning (Topic Summarization)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.3 Analogical Reasoning (FAQ Knowledge Bot)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.4 Workshop Submission

Analogical Reasoning (FAQ Knowledge Bot)

- **Learning Objectives**
  - Understand the procedure of how chatterbot works
  - Learn how to use chatterbot to build a FAQ bot
  - Build a self-learning block to process the question not appear in the dataset
- **Package to use,**
  - Chatterbot 1.0.5 in Python
- **Technique,**
  - Text-based machine reasoning (Frequency-based matching)
- **Case/Scenario,**
  - Autistic Spectrum Disorder FAQ bot
- **Requirement,**
  - Familiar with Python

Before the workshop

Installation & Tutorial:

1. **Type in pip install chatterbot in your anaconda prompt.**
2. **Read documentations about Chatterbot in https://github.com/gunthercox/ChatterBot**

Basic Usage

```
# Create a chatbot called ‘chatbot name’
chatbot = Chatbot(‘chatbot name’)

# Train the chatbot with given corpus
trainer = ChatterBotCorpusTrainer(chatbot)

# You can find more corpuses in the documents provided in https://github.com/gunthercox/ChatterBot
trainer.train(‘chatterbot.corpus.English.greetings’)

# Try greeting with chatbot!
response = chatbot.get_response(‘Hi’)
print(response)
```

Train your chat bot with new conversation (1)

There are several ways to train the chatterbot. ListTrainer is one of them. ListTrainer is used to train the chat bot with a train list. That train list should be in the order of [Q, A, Q, A, Q, A……]

For example:

```
train_list = [
        'Is Sam a good guy?',
        'Certainly yes!',
        'Does Sam wear glasses?',
        'Yes.'
]
```

Train your chat bot with new conversation (2)

```
# Define the trainer and train list
trainer = ListTrainer(chatbot)
train_list = [
    "Is Sam a good guy?",
    "Certainly Yes!",
    "Does Sam wear glasses?",
    "Yes"
]
trainer.train(train_list)

# Let's see what happens
response = chatbot.get_response("Is Sam a good guy?")
print(response)
```

Train an ASD FAQ chat bot

```
# With ListTrainer, we can train an ASD FAQbot with ASD FAQ dataset.
# Similarly, create a ListTrainer
trainer = ListTrainer(chatbot)
# Load the ASD FAQ data and extract the question and answer
data = pd.read_excel('./ASD FAQ KB v001.xlsx', sheet_name='FAQ')
question = data.get('Question')
answer = data.get('Long_Answer')
# Iteratively adding the question and answer
train_list = []
for i in range(len(question)):
    train_list.append(question[i])
    train_list.append(answer[i])

trainer.train(train_list)
```

Workshop

- Open the A1234567X Donald Duck – FAQ Bot python notebook file and complete the workshop
- Build an FAQ bot which is able to learn.
- Requirement: Every time the FAQ bot responses, the user need to provide feedback on whether the answer is good or not.
- If good, FAQ bot learns (stores) the response.
- If not good, user inputs a new answer for the learning bot to learn.

Workshop Submission

- **Naming convention: StudentID YourFullName.sql, e.g. A1234567X Donald Duck – sln – FAQ Bot.ipynb/zip**
- **Use zip to a single file, then rename, if you plan to submit multiple files.**

References

1. **Chatterbot github link:**
   https://github.com/gunthercox/ChatterBot

2.3 Knowledge Representation and Acquisition/Discovery [Workshop]

2.3.1 Deductive Reasoning (Decision Tree)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.2 Inductive Reasoning (Topic Summarization)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

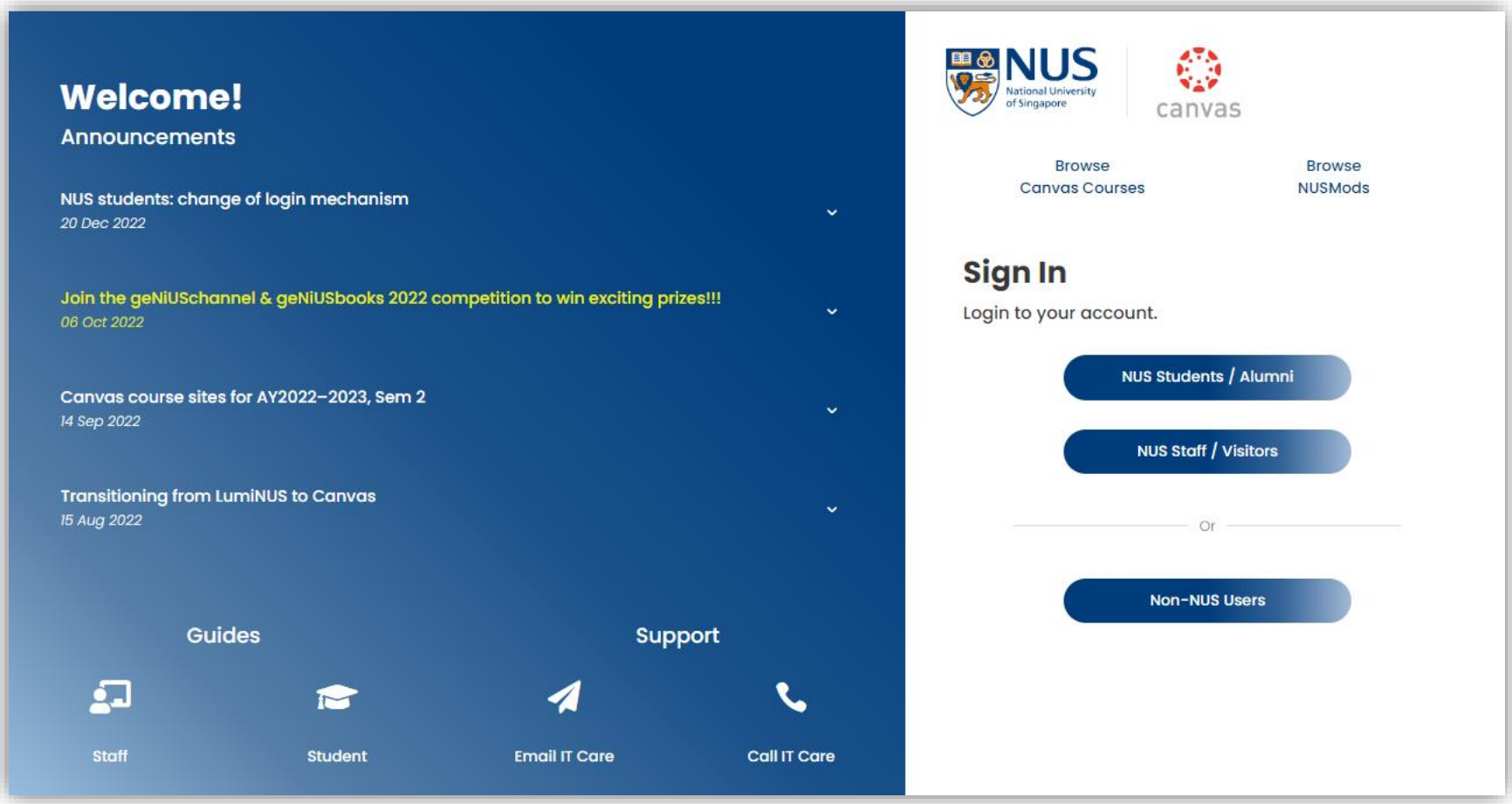
2.3.3 Analogical Reasoning (FAQ Knowledge Bot)

Special thanks to Yan Wei Quan (A0215498U) for his contribution.

2.3.4 Workshop Submission

Workshop Submission

- **Naming convention: StudentID YourFullName**
- **Use zip to a single file, then rename, if you plan to submit multiple files.**

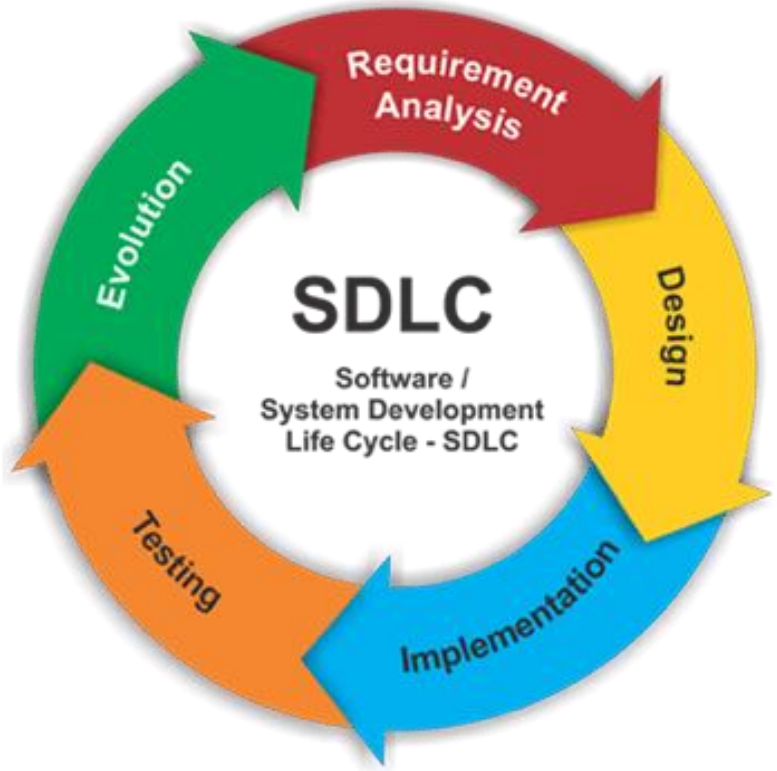


END OF NOTES

APPENDICES

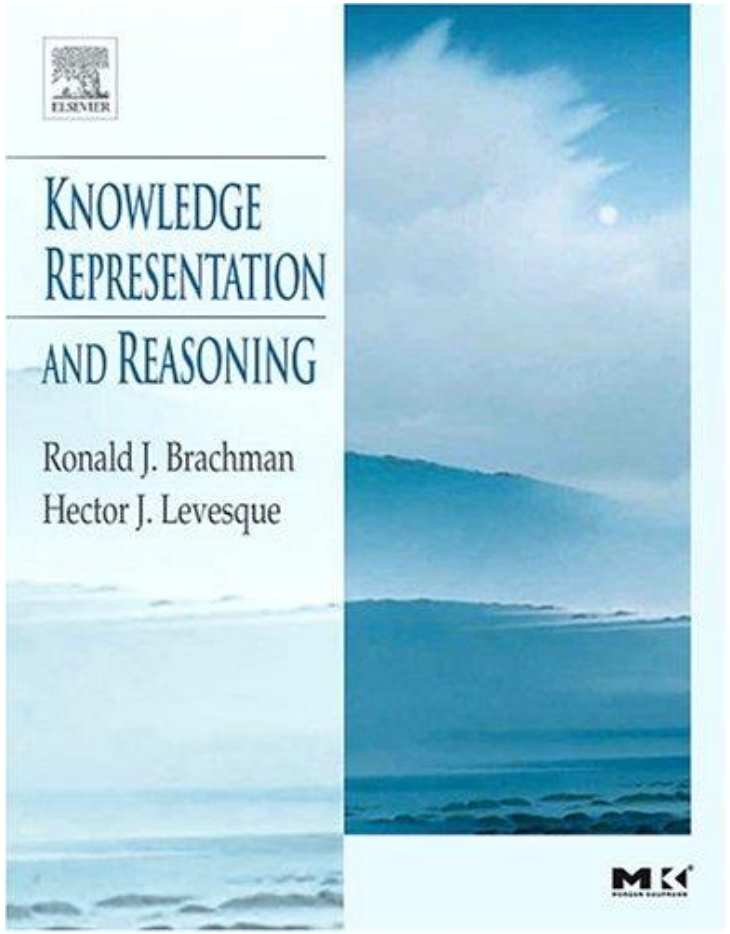
System Development Life Cycle (SDLC)

- **Requirement Analysis**
  - Problem selection: Identify business value and purposes
- **Design (Knowledge Representation and Acquisition)**
  - Knowledge acquisition, interviews
  - Definition of problem domain: Draw high level Inference Diagram
  - System design: Compose other relevant Knowledge Models
- **Implementation (KIE Development)**
  - System development: KIE tools
- **Testing**
  - Integrate, test, revise, deploy, and use
- **Evolution**



https://i1.wp.com/melsatar.blog/wp-content/uploads/2012/03/sdlc.png?fit=830%2C374&ssl=1

Reference



1. Designing a decision service using guided rules

   https://access.redhat.com/documentation/en-us/red_hat_decision_manager/7.2/html-single/designing_a_decision_service_using_guided_rules/

2. KIE Workbench Tutorial : Human Machine Interaction

   https://www.youtube.com/watch?v=NfNnUsr66Cc

3. KIE Workbench Tutorial : Guided Decision Tables

   https://www.youtube.com/watch?v=qBgxVoc2qfw

4. Jay Pujara & Sameer Singh. (2018). Mining Knowledge Graphs from Text

   https://kgtutorial.github.io/

What use cases do our MTech students apply?

How our learners apply the learnt from course:

Challenges

I work in a top **oil & gas** company as a **Pricing Tactics Advisor**. Among the big oil and gas firms and MNCs, our company is exceptionally known for their **rigor in business processes and controls**. For such a huge company, such traditional process and controls can hinder our speed to market, especially in the technology driven world. Many of our processes are **still very manual and excel based**, requiring many levels of endorsement and checks. These processes can be easily replaced with process automation tools.

What learnt is useful?

The **KIE tool** can easily replace many of the existing manual processes that we do. E.g. Email endorsements (sending to the right manager who has the right endorsement authority), request forms via email (many of these email request has missing info and we end up going back and forth. With a validation form, this will not happen).

Decision trees is also one of the relevant modules I have been using in my daily work as the Pricing Tactics Advisor when it comes to dynamic pricing.

How/Where to apply to workplace?

I have actually automated many of the existing **endorsement and email request** using Microsoft Sharepoint (paid software) workflow. It works in a similar concept as KIE.

I have been using JMP (paid software) to make numerous **dynamic/tactical pricing strategies**. Many of these strategies are first developed by exploring our historical competitors and transactional data using python in Jupyter notebook.

Business values

Our company has just begun on their **digital transformation** journey. I believe that the company of tomorrow is not one who has the most advanced technology, but the company who has the most amount of data. Many of the oil and gas firms are sleeping giants. We have a treasure trove of untapped data. It is a journey for us to move away from our oil and gas mind-set into the world of digital technology. I am leading the Asia Pacific Market Entry Strategy for China and we are **exploring new and lean ways** to do market entries without the baggage of traditional systems that the mothership is using.

How our learners apply the learnt from course:

Challenges

I build **credit risk** systems for the **bank**. The difficult part is:

1. There are huge number of credit policies, they **scattered around in many places**: within systems, excels, or in the human brains of those highly experienced account managers, they are **not synchronized**.

2. **Polices keep on changing** due to policies changing, regulatory requirements, etc. Some policies were built in the system and **logic were coded in programs**. It took very long time to adjust them as standard system development life cycle SDLC kicks in.

3. There is no centralized knowledge base which serve as the golden source of rules, there is also no centralized data taxonomy, which causes **conflicting results** and no one can tells which one is correct.

What learnt is useful?

The **KIE suite** is very useful in terms of **building and testing the policies** in the bank credit departments.

The **business users** could use the graphic tools to come up the flow, input their rules as guided decision table, auto generate the forms if input is required, and quickly start their testing on the new policies. (**rather than wait for tech team to finish the coding and test**)

For tech team, we could either code the tested policies into the legacy systems, or use the KIE suite to **expose the policies/rules into application programming interface API**, and simply call it.

This will significantly **shorten the turn around time** for new policies launch.

How/Where to apply to workplace?

Take the policies as example, we could easily **de-couple** the coding part and the logic part between development team and business team.

The business team could focus on the policies and come up the list of **mutually agreeable rules table in excel sheet**, track them in version management tools such git, to make all the rules traceable, and prevent multiple conflicting version.

Whenever there is need to change rules, the credit system will **simply load the excels and the polices are live**. **No software deployment** is required, this will avoid the software bugs which happens quite often for typical code deployment.

Business values

It **reduces the time** cost of launching new policies: as it will be shorter development time, shorter testing and deployment time.

It **decouples** the business logic and technical details, **easier for maintenance** and future system migration, which again saving cost.

How our learners apply the learnt from course:

Challenges

I am a **robotic process automation RPA** developer working for a major **bank**, and I am developing software robots to aid bank staff in automating tedious, but noncomplex tasks. There are few phases in the software development life cycle, and one of the difficulty we face is most likely **finalizing the business requirements**.

Another difficulty would be the business as usual BAU operations after development. Although we told the users to put in a specific file format (**RPA is not intelligent if we are looking at low cost solutions**), they will put in different formats causing the robot to fail. Hence we spend a bulk of time in development **writing exceptions** to prevent these.

What learnt is useful?

Firstly, this course has taught me how to use **KIE tool** (jBPM/Drools). This will definitely aid me in my development in the bank because I use a similar vendor product (Kofax Totalagility [KTA]). Although we use KTA to do basic BPM, we **do not integrate any intelligence** when using it, since most development time is short and they take man-hours budget into heavy consideration. So from this course I learnt to **integrate business process driven by a** (knowledge driven) **rule engine**. Also, by learning different techniques of reasoning, it will be helpful as a tool to help sketch out **multiple methods and models in deriving a smarter robot**. For example, since we deal with **lots of exceptions**, we can use a rule-based reasoning engine (for example, **guided decision table**) to resolve it to different scenarios, which would be **efficient** than having to write a line of code for every decision.

How/Where to apply to workplace?

We currently process company documents in one of our AML (Anti-Money Laundering) robots, which actively seeks for sanction words (example: nuclear) using OCR, which will be used to approve/reject loans by a customer. Right now, we have a process maker (operations staff), and a **checker (management)**. If a rule-based engine is well designed, it can even replace the checker. But a full replacement will not be recommended, and the checker should do double check. But this will **speed up document processing** in a day, improving efficiency.

Business values

Higher efficiency can be reached. In RPA we count profits by total hours saved. By introducing **more reliable and intelligent agents** in our robots, besides from human action assistance, we can replace human thinking cognition, thereby saving even more man hours.

How our learners apply the learnt from course:

Challenges

I am **test engineer** in **semiconductor** company that produce memory chips. I face difficulty in **identifying the failure mode of the memory chip** that I am testing. I will need to **forecast the yield for each product** and each product is having different attribute or properties. I am having hard time to process all the yield data and finally produce the yield forecast. I also need to generate test time forecast at the same time. As a result, I will need better forecast model.

What learnt is useful?

I can use Python **Orange tool** for **data mining**. I can key in the testing result in **excel form** then use the decision tree to find out which attribute/property of the testing parts **affects the testing results the most**.

KIE workbench tools can help me in creating **GUI**. Enable **multiple users** to key in their input data into the platform. They can also retrieve output that there are interested. They can also send their question using the KIE forms.

How/Where to apply to workplace?

I will use python orange for data mining. Finding out the **best decision tree then use it to create rules** that can **forecast the yield** of each product.

Then create KIE workbench application for **yield prediction**. **Multiple user** can key in the product yield value with properties they have. Then, the rule generated using data mining method can be applied for forecast.

Business values

I can help my company to **enhance** its yield forecast system and have a better production volume forecast. This will be chance to use system intelligence for cost saving.

I can also prepare myself in data mining and implementing rule based system in my working environment. Enhance my own skill set.

How our learners apply the learnt from course:

Challenges

I work in a **Systems Integration** company. Here we have multiple large-scale projects which require customization, development and implementation. It is difficult to manage scheduling and assignment of development effort to projects (due to skillsets, availability, etc.).

Also, as the company has been around for quite a while, much of the processes are still **manual and archaic**. As people come and go, the initial processes are no longer adhered to and there is **not much visibility** on them.

What learnt is useful?

KIE tool can encode the various **decision factors** for certain processes like (leave, scheduling of staff to projects). It also allows for **multiple users** to interact and utilize the system at once.

For example, in my company there are many projects and developers. KIE can help to automate the assignment, monitoring and scheduling of developers to projects.

How/Where to apply to workplace?

First, it makes sense to **map out the current given process** within my company. For example, I've recently done a Business Process Re-engineering to improve the Recruitment Process in my company. Using KIE, I can map out the process and include the business rules/ logic that is used to decide.

Secondly, **include the business logic** into the Recruitment Process. For example, the salary approval for certain staff rank must be routed and go through the Head of HR or to the Head of the Business Unit. This can be mapped within KIE and sent to them for approval.

Lastly, on-board users to utilize this automated process. KIE can be used as the system in which to **implement the automated process**.

Business values

One business value that can be derived is to **decrease** the required manual effort for writing out all the **physical forms**. Another is to **visualize** and provide awareness of the current **process**. A third is to ensure that the process is **fair** and not biased.

How our learners apply the learnt from course:

Challenges

I worked in a **charity** called Singapore Children's Society. My work involves **conducting research** on issues concerning the well-being of children, youth and family e.g., child abuse and bullying. One of the main work challenges in my field of work is **translating scientific findings into practices that social workers can then use to help their clients e.g., children and youths.**

What learnt is useful?

The most helpful thing that I had learnt from this course is **how to take the acquired knowledge that I had gained from my own research and from other's research findings and convert them into rules.** For example, I have converted these research findings into decision trees knowledge model and rules that I then give to social workers **to assist their decision-making**.

The **KIE tool** can support better to the social workers because it can be automated thus I do not need to sit down with social workers every time there is a change in the model. KIE can be scaled up to **support multiple users**, which help us a lot because **we are a small team with little manpower**. The other thing is that KIE can **generate reports easily**, which makes it easy to report to management on **how things are progressing** without having to spend time doing up reports. This leave us with more time to take on more higher value work.

How/Where to apply to workplace?

I used KIE to generate a small project based on the decision tree that ask the social workers to fill in the inputs in order to generate a set of recommendation on how they can work with their clients. For example, my research investigated whether certain demographic profile of children was more at risk of child abuse. I looked at variables such as race, age, gender, parental education level and examine the association with child abuse victimization. **I then use these findings to build a decision tree to derive simple rules to fit into the KIE project/system**.

Business values

Because of the course, I had learnt to implement a **automated machine reasoning systems** that did part of my work for me. The business values derived from this is that my knowledge is now readily available to social workers, **knowledge can be updated easily**, social workers is now better able to use the knowledge because it has been translated into easily applied rules, and I can focus more on **higher values work** such as doing more research.

Knowledge Representation and Acquisition/Discovery using Decision Tree

Rule Induction using Decision Tree

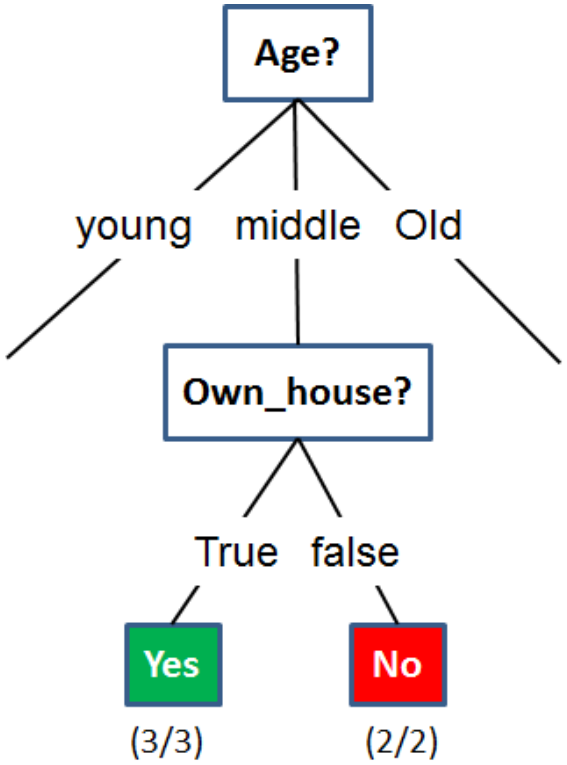
Bank Loan Example – Business Background

- Banks receive many loan applications that has to be assessed for approval.
- Each application consists of many factors such as **Age, Job status, Housing, Credit history**.
- Some applications are approved, others are not; Some debtors default, others don't.
- Banks dislike defaulters. Banks want to approve only applicants who are unlikely to default.
- Bank's task is to predict if a new applicant will default or not.
- This a classification problem: **Approve** projected non defaulter or **Reject** projected defaulter during loan application.

Rule Induction using Decision Tree

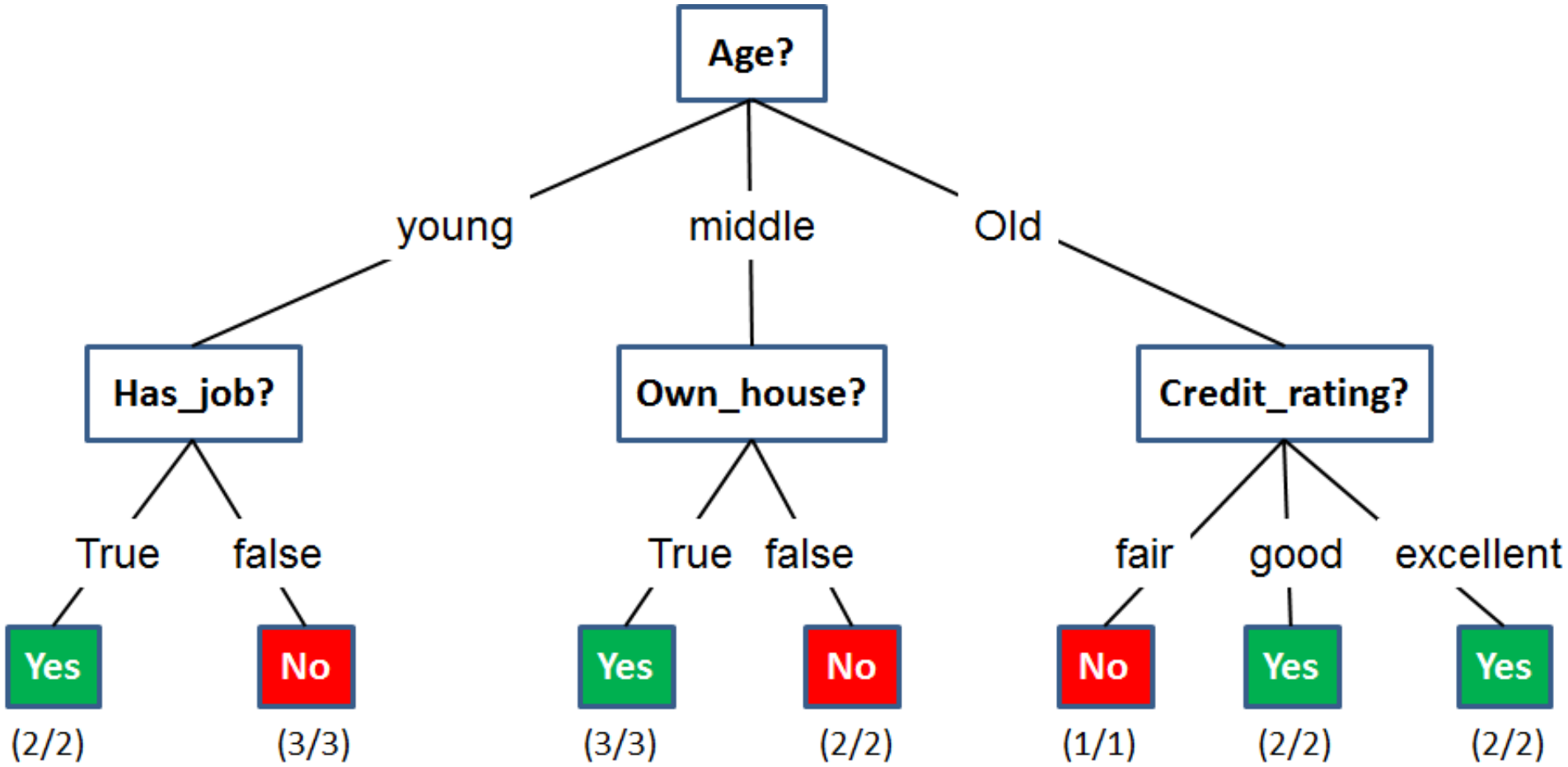
Bank Loan Example – Rule Induction Data Science

ID	Age	Has_job	Own_house	Credit_rating	Outcome
1	young	False	False	fair	No
2	young	False	False	good	No
3	young	True	False	good	Yes
4	young	True	True	fair	Yes
5	young	False	False	fair	No
6	middle	False	False	fair	No
7	middle	False	False	good	No
8	middle	True	True	good	Yes
9	middle	False	True	excellent	Yes
10	middle	False	True	excellent	Yes
11	old	False	True	excellent	Yes
12	old	False	True	good	Yes
13	old	True	False	good	Yes
14	old	True	False	excellent	Yes
15	old	False	False	fair	No



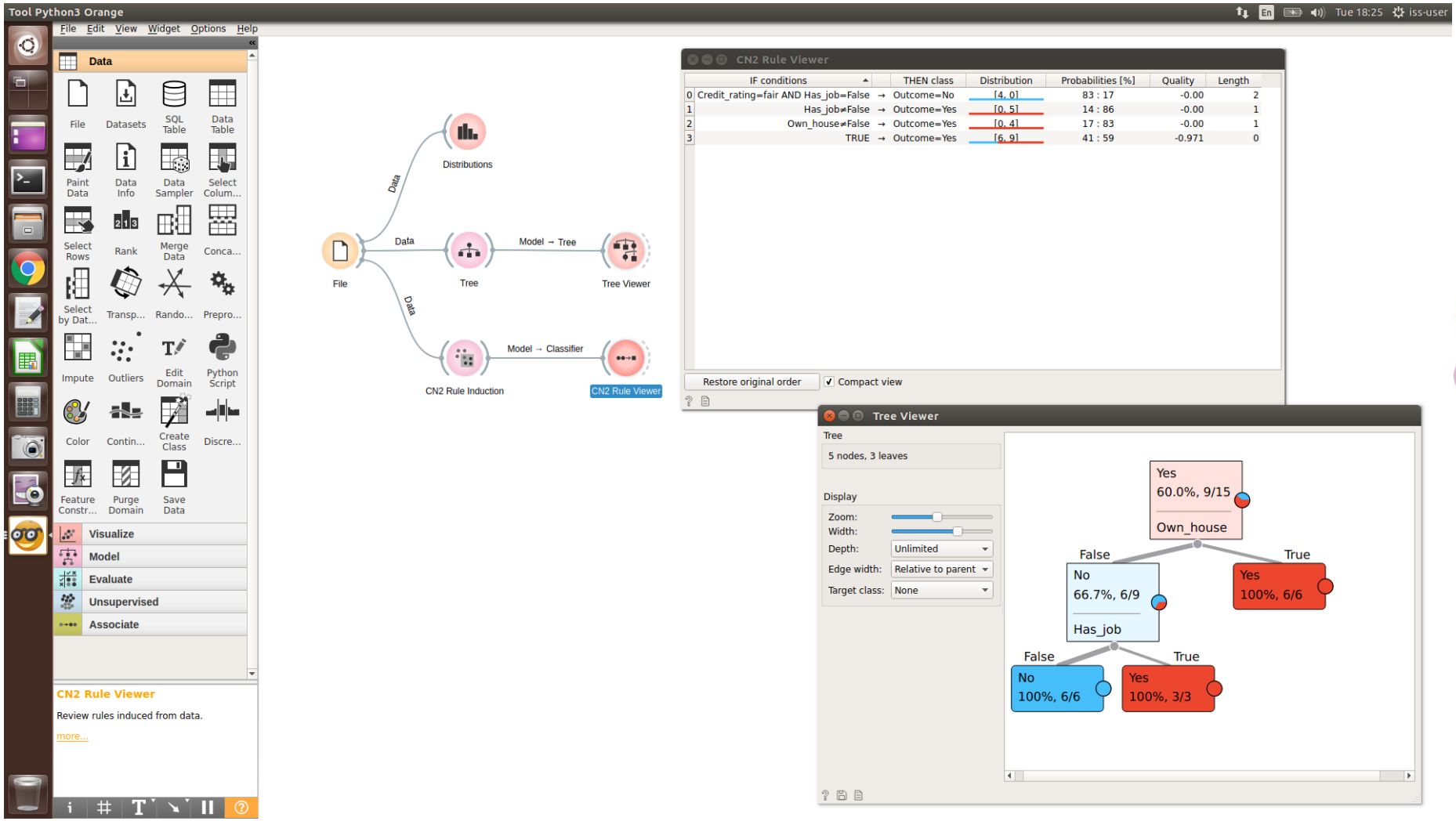
Rule Induction using Decision Tree

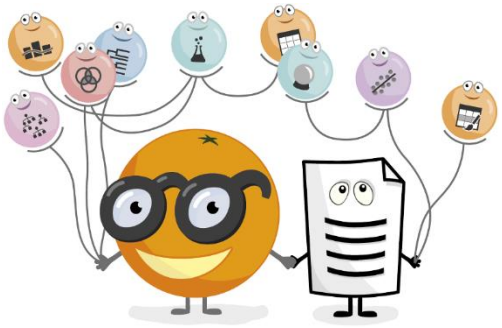
Bank Loan Example – Decision Tree



Rule Induction using Decision Tree

Data Mining Tool: Orange3 (python)

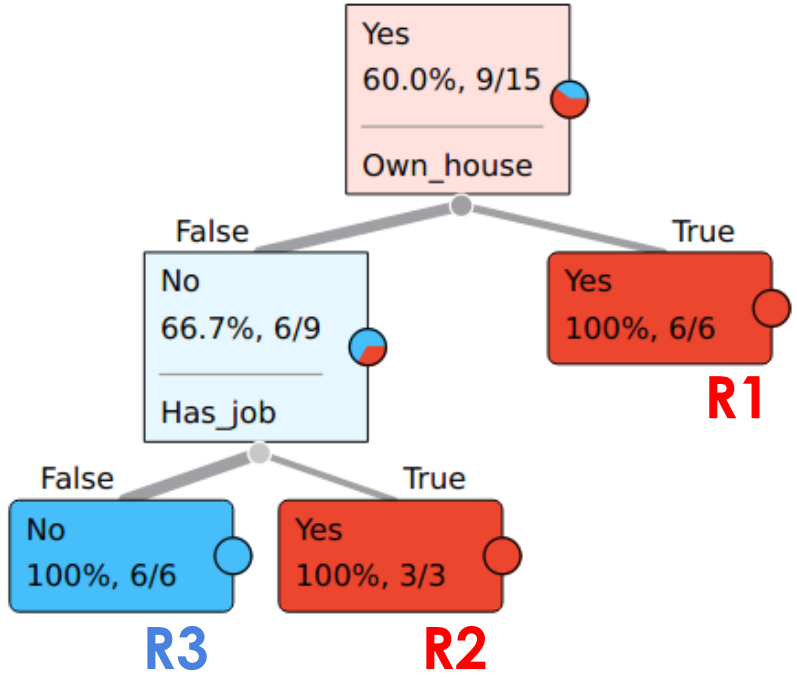




Rule Induction using Decision Tree

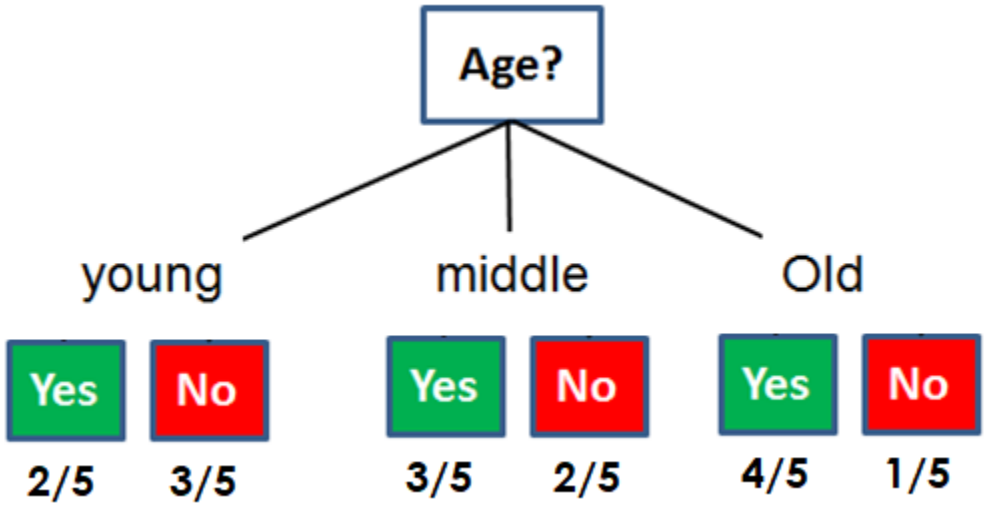
Orange3 Bank Loan Example – Decision Tree

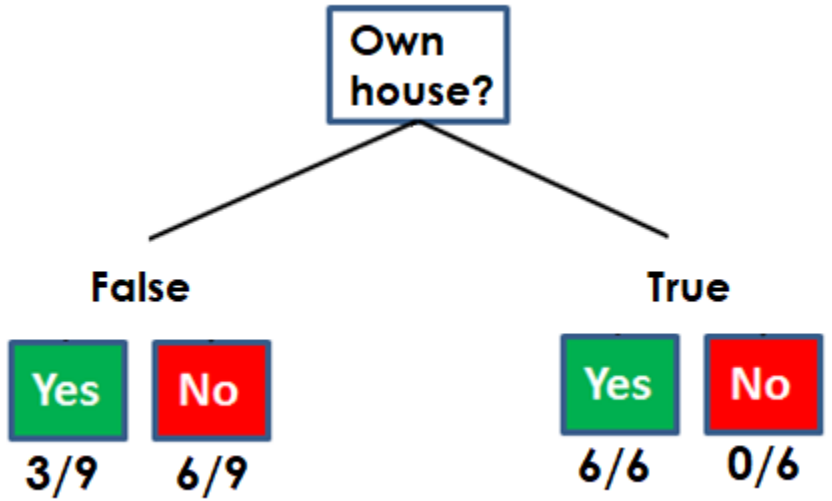
ID	Age	Has_job	Own_house	Credit_rating	Outcome	
1	young	False	False	fair	No	R3
2	young	False	False	good	No	
3	young	True	False	good	Yes	R2
4	young	True	True	fair	Yes	R1
5	young	False	False	fair	No	
6	middle	False	False	fair	No	R3
7	middle	False	False	good	No	
8	middle	True	True	good	Yes	
9	middle	False	True	excellent	Yes	
10	middle	False	True	excellent	Yes	R1
11	old	False	True	excellent	Yes	
12	old	False	True	good	Yes	
13	old	True	False	good	Yes	
14	old	True	False	excellent	Yes	R2
15	old	False	False	fair	No	R3



Rule Induction using Decision Tree

Decision Tree Algorithm – Which feature to select for split?





- **Which attribute is more intuitively for your to better/easier decision making?**

Rule Induction using Decision Tree

Decision Tree Algorithm – ID3 Information Gain

ID3 algorithm

- **In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset.**

- **Information Gain formula:**

  IG(DataSubsets|Attribute) = Imp(Initial Dataset) – Imp(DataSubsets|Attribute)

Rule Induction using Decision Tree

Decision Tree Algorithm – Initial Dataset Impurity

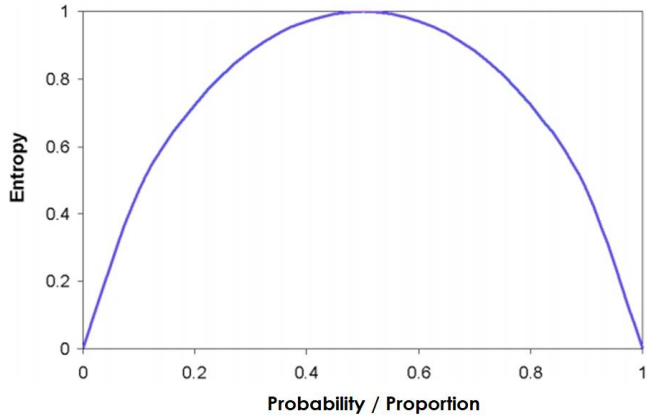
ID	Age	Has_job	Own_house	Credit_rating	Outcome
1	young	False	False	fair	No
2	young	False	False	good	No
3	young	True	False	good	Yes
4	young	True	True	fair	Yes
5	young	False	False	fair	No
6	middle	False	False	fair	No
7	middle	False	False	good	No
8	middle	True	True	good	Yes
9	middle	False	True	excellent	Yes
10	middle	False	True	excellent	Yes
11	old	False	True	excellent	Yes
12	old	False	True	good	Yes
13	old	True	False	good	Yes
14	old	True	False	excellent	Yes
15	old	False	False	fair	No

Yes | No

9 | 6

$P_1(Yes) = 9/15$

$P_2(No) = 6/15$

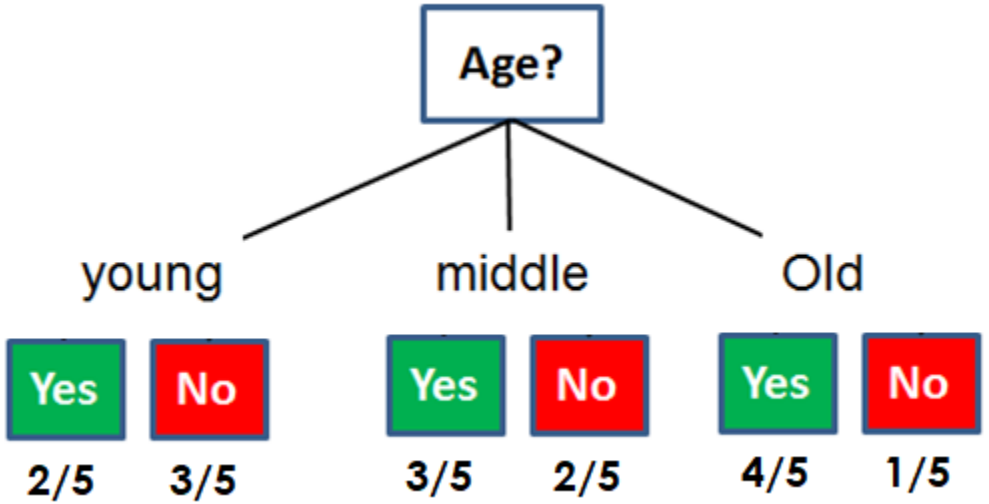


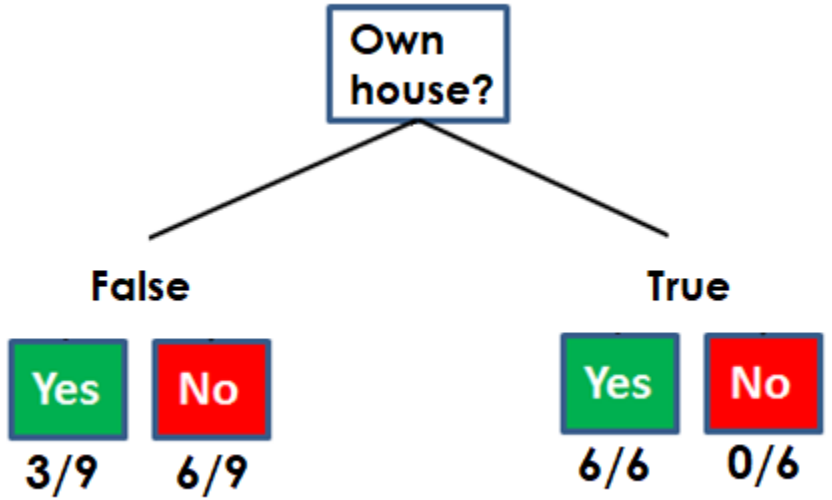
$$Imp(D_j) = Entropy(p) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

Imp(Initial Dataset) = - 9/15 log(9/15) - 6/15 log(6/15) = 0.5288 + 0.442 = 0.971

Rule Induction using Decision Tree

Decision Tree Algorithm – Data Subset Impurity by attribute split



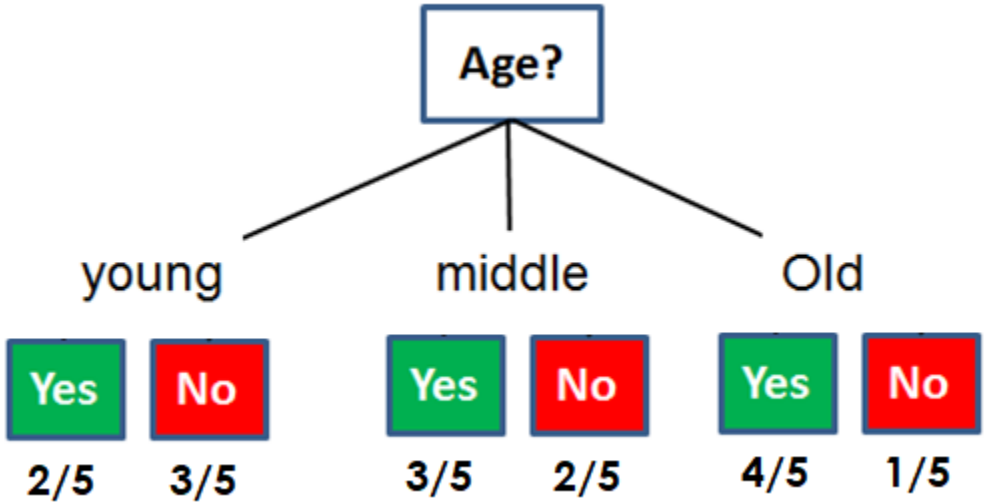


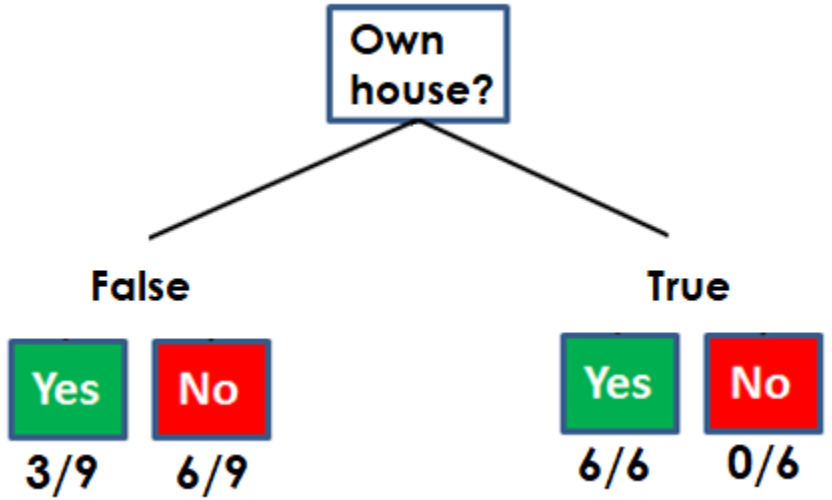
$$Imp(D_j) = Entropy(p) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

$$Imp(\{D_1, \ldots, D_l\}) = \sum_{j=1}^{l} \frac{|D_j|}{|D|} Imp(D_j)$$

Rule Induction using Decision Tree

Decision Tree Algorithm – Which attribute for conditional split?





Imp(Young) =

-2/5log(2/5)-3/5log(3/5)

= 0.5288 + 0.442

= 0.971

Imp(Middle)

= 0.971

Imp(Old) =

-4/5log(4/5)-1/5log(1/5)

= 0.2575 + 0.4644

= 0.722

Total Imp(Age) = 5/15 x 0.971 + 5/15 x 0.971 + 5/15 x 0.722

= 0.3237 + 0.3237 + 0.2407 = 0.888

IG(D | Age) = Imp(Initial Dataset) - Imp(D | Age)

= 0.971 – 0.888 = 0.083

Imp(False) =

-3/9log(3/9)-6/9log(6/9)

= 0.5283 + 0.39

= 0.918

Imp(True) =

-6/6log(6/6)-0/6log(0/6)

= 0 + 0

= 0

Total Imp(Own_house) = 9/15 x 0.918 + 6/15 x 0

= 0.551 + 0 = 0.551

IG(D | Own_house) = Imp(Initial Dataset) - Imp(D | Own_house)

= 0.971 – 0.551 = 0.42 ☺ The larger IG, the better attribute split !

Rule Induction using Decision Tree IDENTIFY ALIENS

Aliens

Not aliens

Training Data

SN	Triangle	Antenna	Teeth	Eyes	Alien
1	1	3	1	2	TRUE
2	1	3	0	2	TRUE
3	1	3	1	2	TRUE
4	1	3	0	3	TRUE
5	1	2	1	2	FALSE
6	0	3	0	3	FALSE
7	1	6	0	2	FALSE
8	0	3	0	2	FALSE

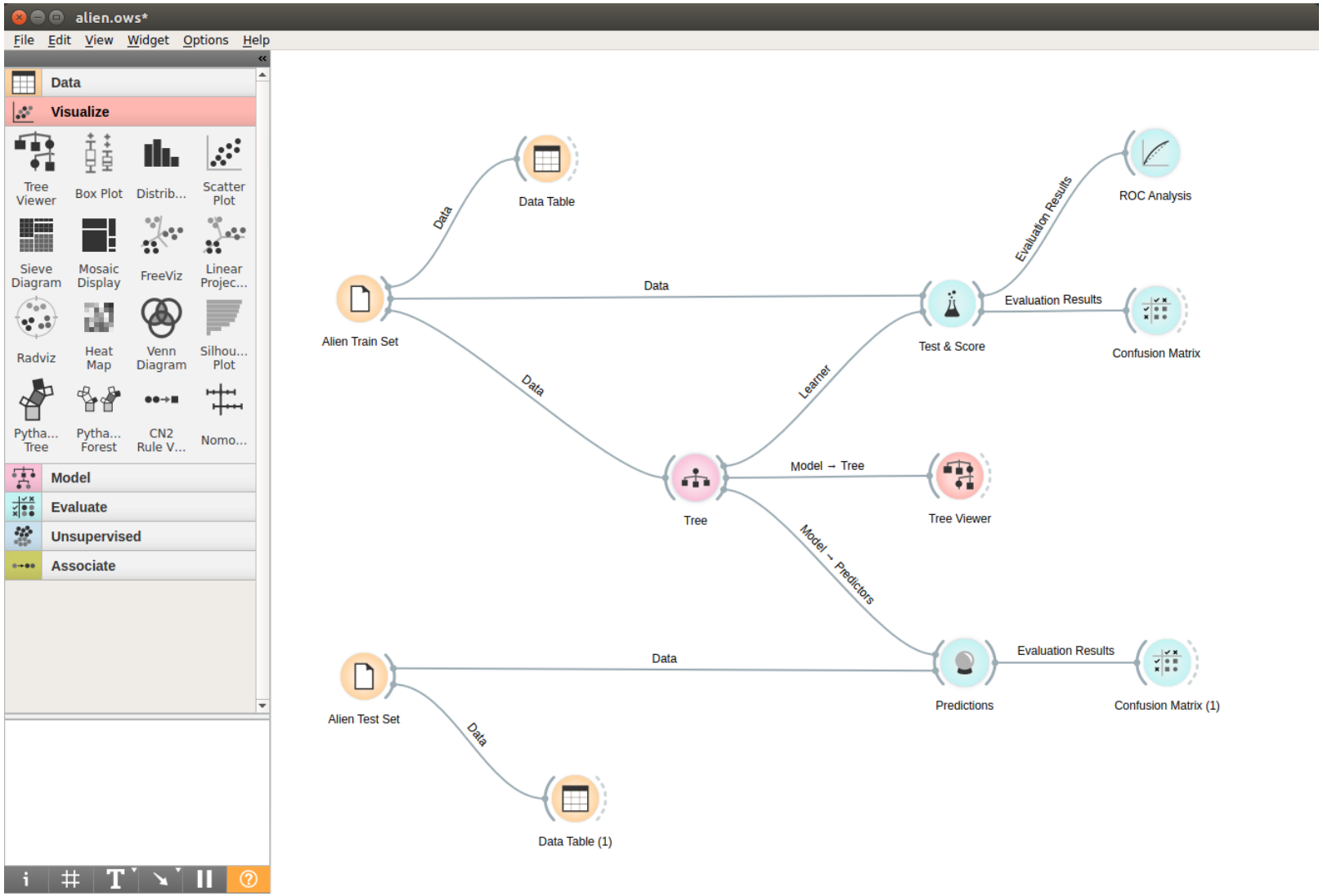
Which one is alien?

A B C D E

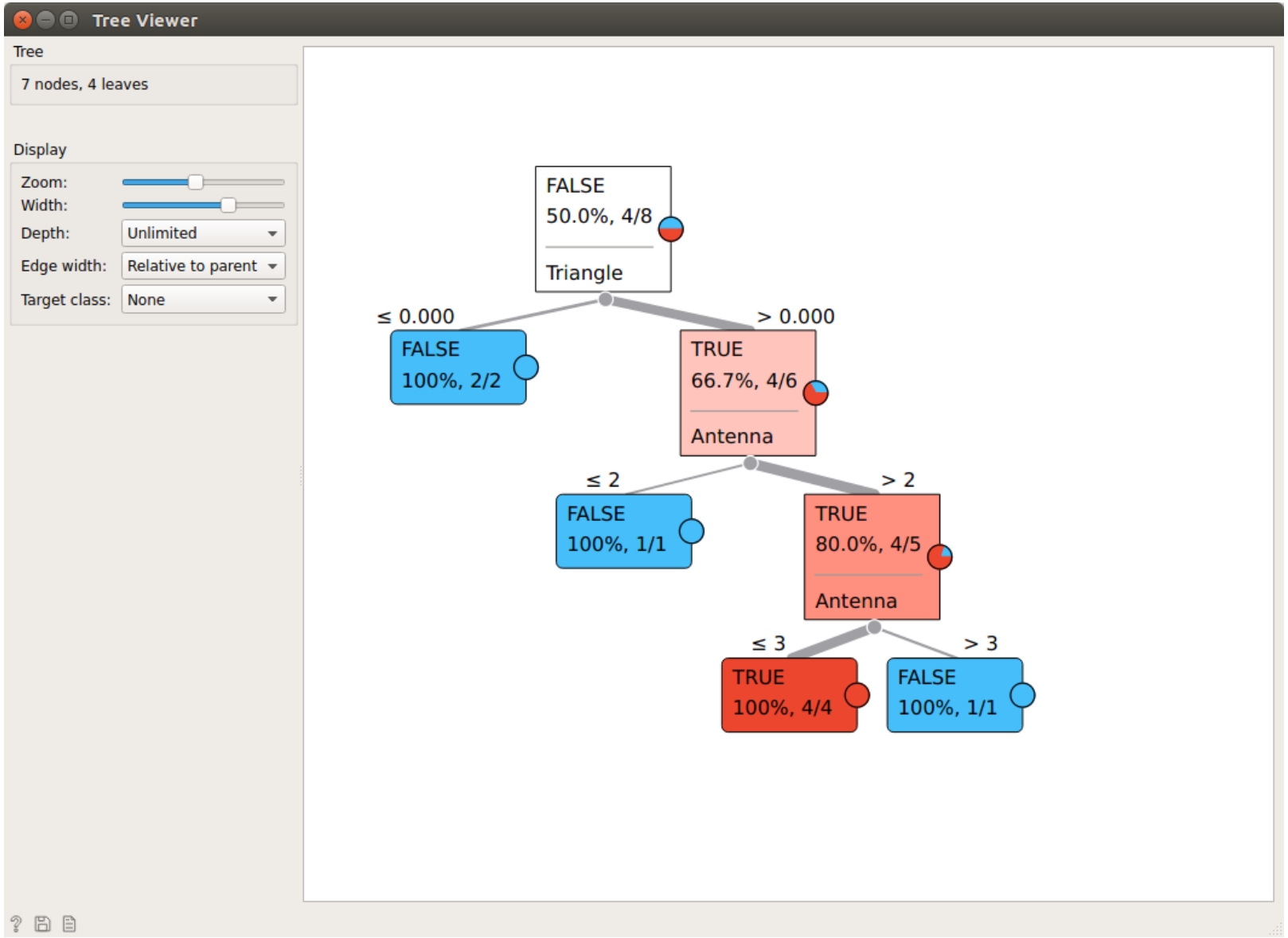
Test Data

SN	Triangle	Antenna	Teeth	Eyes	Alien
A	1	2	0	2	FALSE
B	3	2	1	2	FALSE
C	1	4	0	2	FALSE
D	1	3	0	2	TRUE
E	0	3	0	2	FALSE

Rule Induction using Decision Tree IDENTIFY ALIENS



Rule Induction using Decision Tree IDENTIFY ALIENS



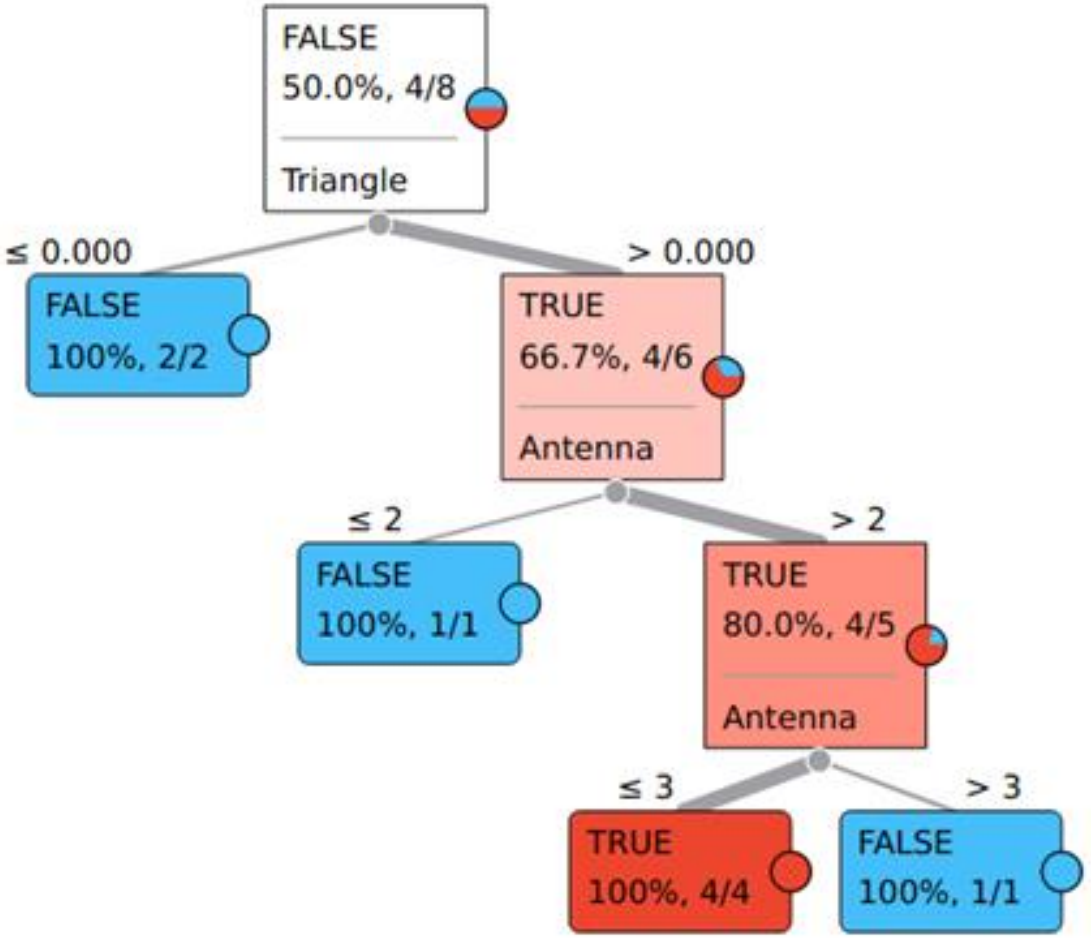
Rule Induction using Decision Tree IDENTIFY ALIENS

Aliens

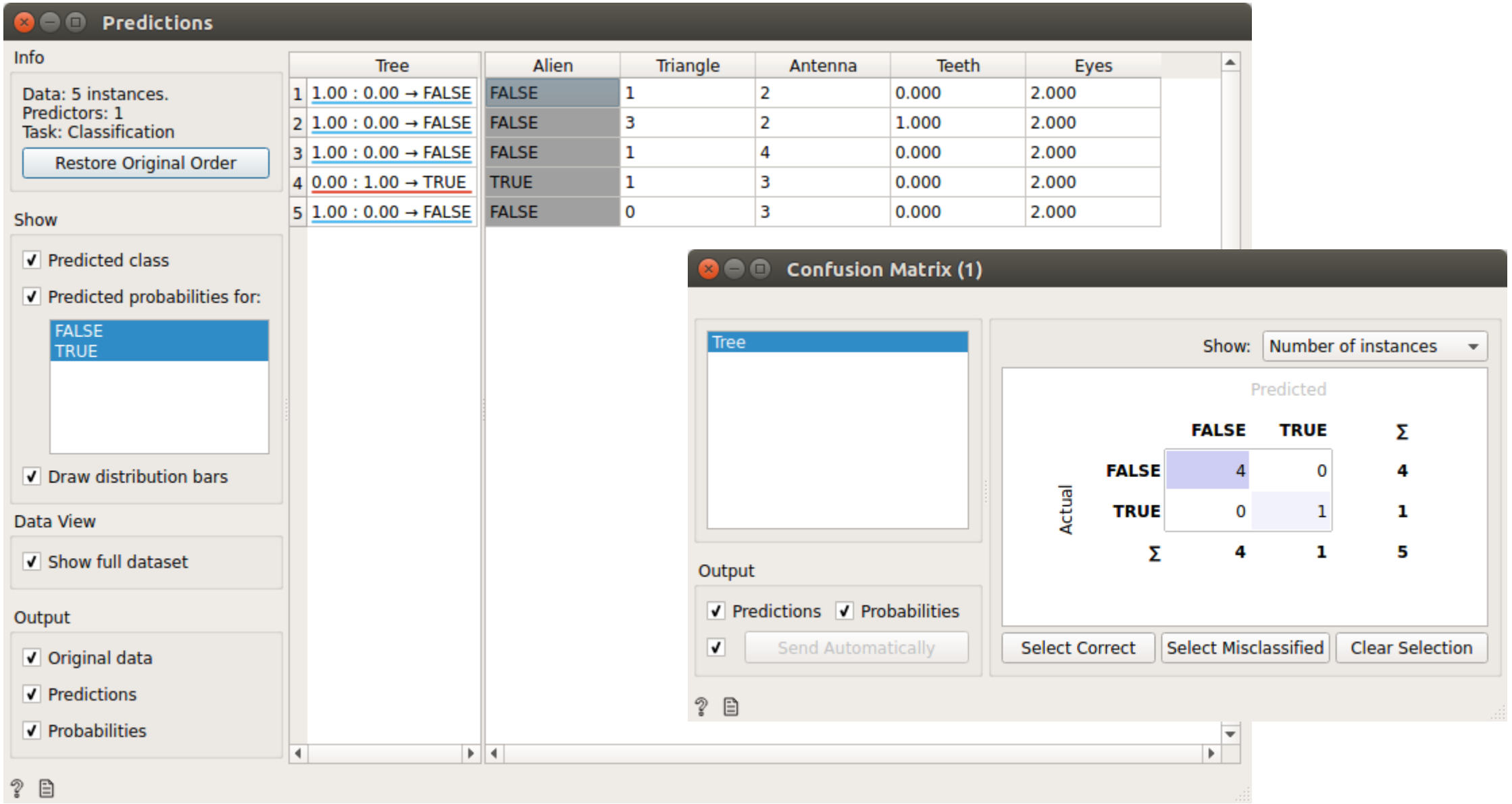
Not aliens

Which one is alien?

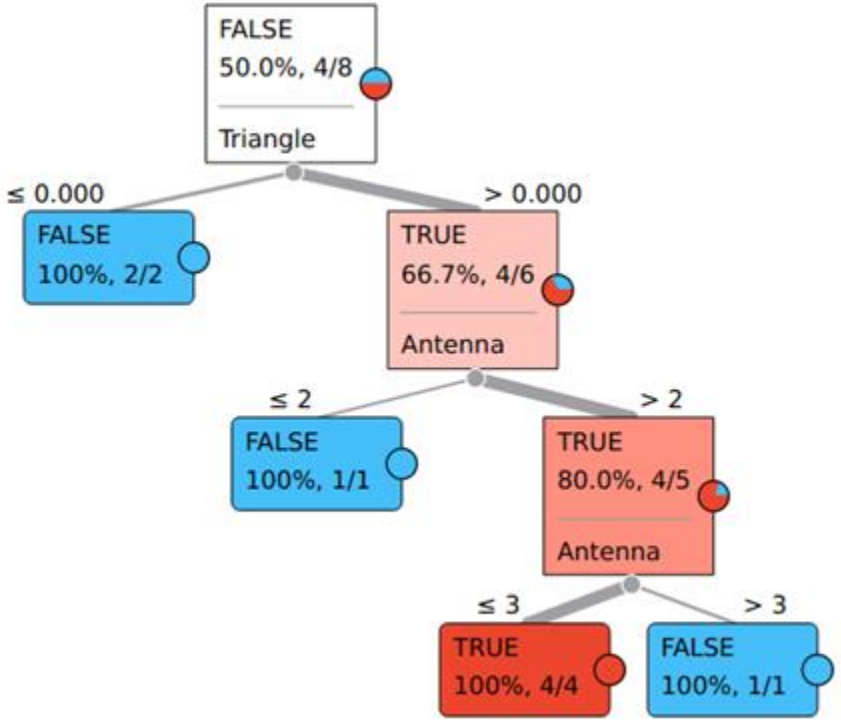
A B C D E



Rule Induction using Decision Tree IDENTIFY ALIENS



Rule Induction using Decision Tree IDENTIFY ALIENS



Test Data

SN	Triangle	Antenna	Teeth	Eyes	Alien
A	1	2	0	2	FALSE
B	3	2	1	2	FALSE
C	1	4	0	2	FALSE
D	1	3	0	2	TRUE
E	0	3	0	2	FALSE
F	2	3	0	2	FALSE

Actual \ Predicted	FALSE	TRUE	Σ
FALSE	4	1	5
TRUE	0	1	1
Σ	4	2	6

	Tree	Alien	Triangle	Antenna	Teeth	Eyes
1	1.00 : 0.00 → FALSE	FALSE	1	2	0.000	2.000
2	1.00 : 0.00 → FALSE	FALSE	3	2	1.000	2.000
3	1.00 : 0.00 → FALSE	FALSE	1	4	0.000	2.000
4	0.00 : 1.00 → TRUE	TRUE	1	3	0.000	2.000
5	1.00 : 0.00 → FALSE	FALSE	0	3	0.000	2.000
6	0.00 : 1.00 → TRUE	FALSE	2	3	0.000	2.000

Which one is alien?

A **B** **C** **D** **E**

F

The 'black swan': unseen before in historical data (no scenario/representation in training data)

END OF APPENDICES