



CSCE 580: Introduction to AI

Lectures 3 and 4: Introduction to AI, Trust and Real-World Applications

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

26TH AND 30TH AUG 2025

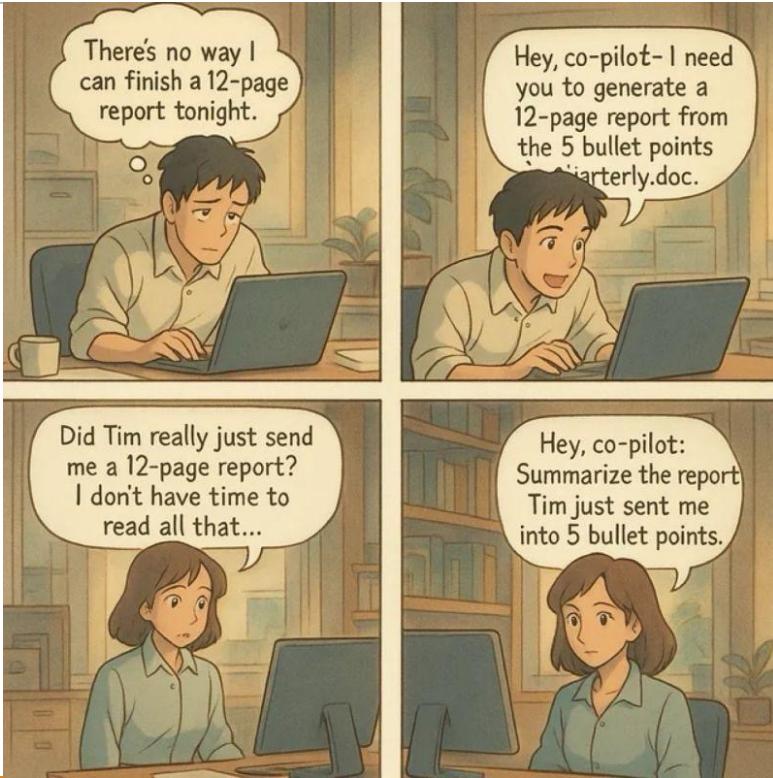
Carolinian Creed: “I will practice personal and academic integrity.”

Credits: Copyrights of all material reused acknowledged

Organization of Lectures 3, 4

- Introduction Section
 - Recap from Week 1 (Lectures 1 and 2)
 - Project A: Q/A
 - AI news
- Main Section
 - L3: Data Prep, Knowledge Graph
 - L4: ML Basics
- Concluding Section
 - About next week – Lectures 5, 6
 - Ask me anything

Introduction Section



Credit: From Internet

Recap from Week 1

- We talked about
 - AI - basics
 - Course logistics
 - Data
 - Chatbots
 - AI ethics
 - Project A starts
- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
 - Weeks 2: Data: Formats, Representation, ML Basics
 - Week 3: Machine Learning – Supervised (Classification)
 - Week 4: Machine Learning - Unsupervised (Clustering) –
 - Topic 5: Learning neural network, deep learning, Adversarial attacks
 - Week 6: Large Language Models – Representation and Usage issues
 - Weeks 7-8: Search, Heuristics - Decision Making
 - Week 9: Constraints, Optimization – Decision Making
 - Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
 - Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
 - Week 13: Trustworthy Decision Making: Explanation, AI testing
 - Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Projects A: Start (4 weeks; 200 points)

- End date: **Thursday, Sep 18**
- Model AI Assignments: <http://modelai.gettysburg.edu/>
 - Choose a project, preferably within last 5 years (i.e., after 2020).
 - Enter its name in “Student-InfoShared ..” sheet, column G
 - Follow instructions and do it alone
 - Submit project outcome
 - Create a project on your Github called **ProjectA**.
 - Put deliverables, as per project description, inside the folder, and commit.
 - Timestamp will be used to confirm that Project-A is done on time

AI News

PROGRAMMING WITH AN AI ASSISTANT



Credit: From FB

MIT Study: 95% GenAI Investments with No Return

- Report: [https://mlq.ai/media/quarterly_decks/v0.1 State of AI in Business 2025 Report.pdf](https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf)
 - Press: <https://www.forbes.com/sites/andrehill/2025/08/21/why-95-of-ai-pilots-fail-and-what-business-leaders-should-do-instead/>

Key points

- “\$30–40 billion in enterprise investment into GenAI, this report uncovers a surprising result in that 95% of organizations are getting zero return.”
- “High Adoption, Low Transformation”

Industry	Key Signals
Technology	New challengers gaining ground (e.g., Cursor vs Copilot); shifts in workflows
Media & Telecom	Rise of AI-native content; shifting ad dynamics; incumbents still growing
Professional Services	Efficiency gains; client delivery remains largely unchanged
Healthcare & Pharma	Documentation/transcription pilots; clinical models unchanged
Consumer & Retail	Support automation; limited impact on loyalty or leaders
Financial Services	Backend automation; customer relationships stable
Advanced Industries	Maintenance pilots; no major supply chain shifts
Energy & Materials	Near-zero adoption; minimal experimentation

Five Myths about GenAI in Business

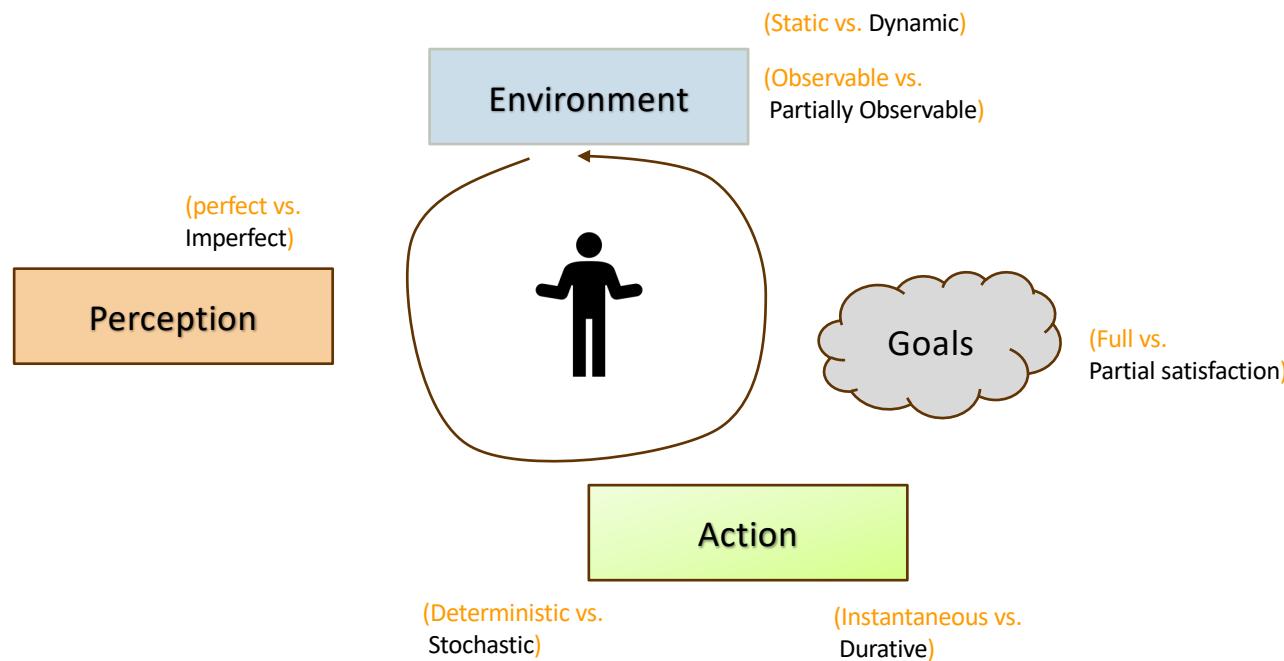
1. AI Will Replace Most Jobs in the Next Few Years → Research found **limited layoffs from GenAI, and only in industries that are already affected significantly by AI**. There is no consensus among executives as to hiring levels over the next 3-5 years.
2. Generative AI is Transforming Business → Adoption is high, but transformation is rare. Only **5% of enterprises have AI tools integrated in workflows at scale and 7 of 9 sectors show no real structural change**.
3. Enterprises are slow in adopting new tech → Enterprises are **extremely eager to adopt AI and 90% have seriously explored buying an AI solution**.
4. The biggest thing holding back AI is model quality, legal, data, risk → **What's really holding it back is that most AI tools don't learn and don't integrate well into workflows**.
5. The best enterprises are building their own tools → **Internal builds fail twice as often**.

Source: https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf

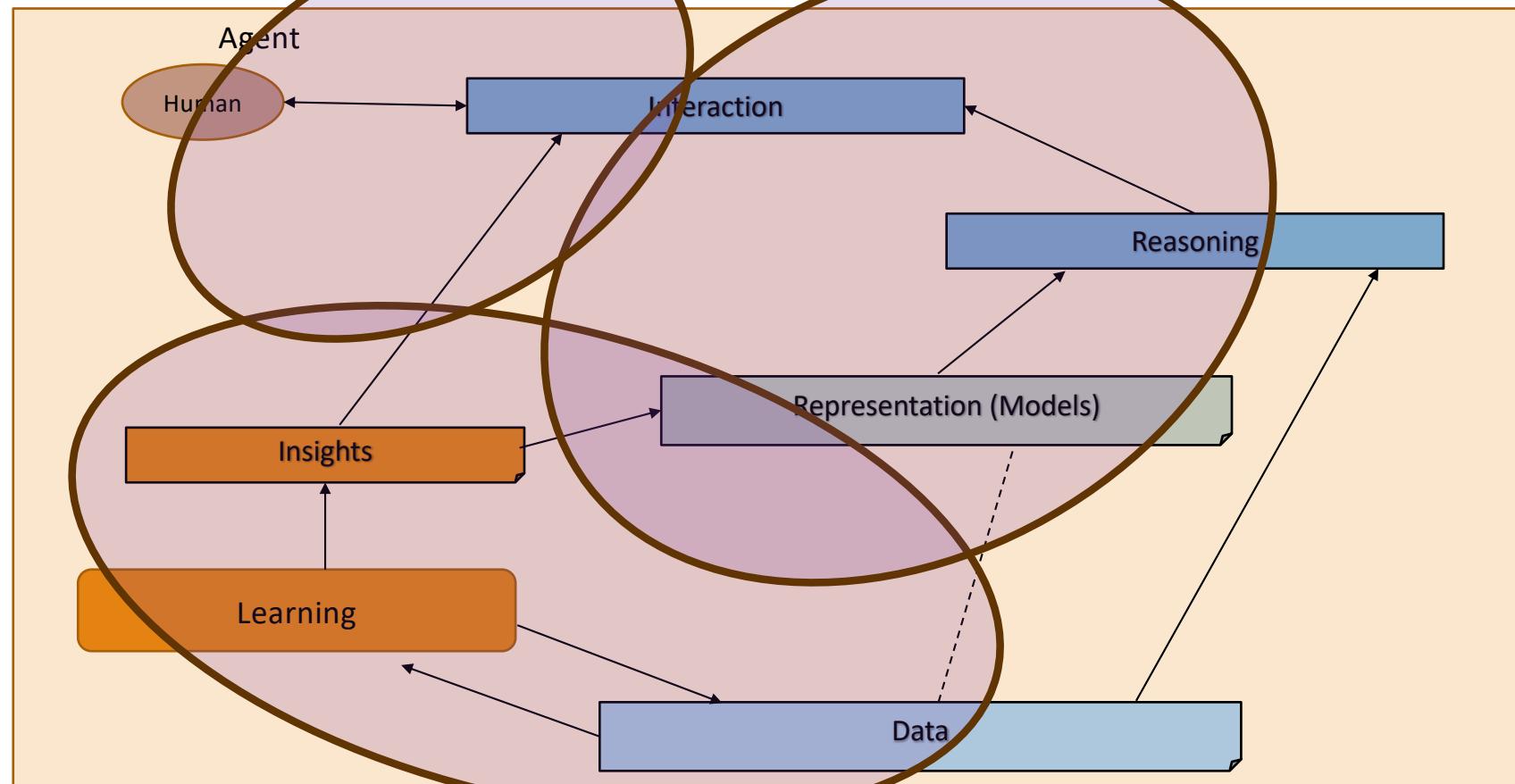
Lecture 3: Representing and Organizing Data

- Data preparation
- Knowledge representation/ graph
- Ontology

Intelligent Agent Model

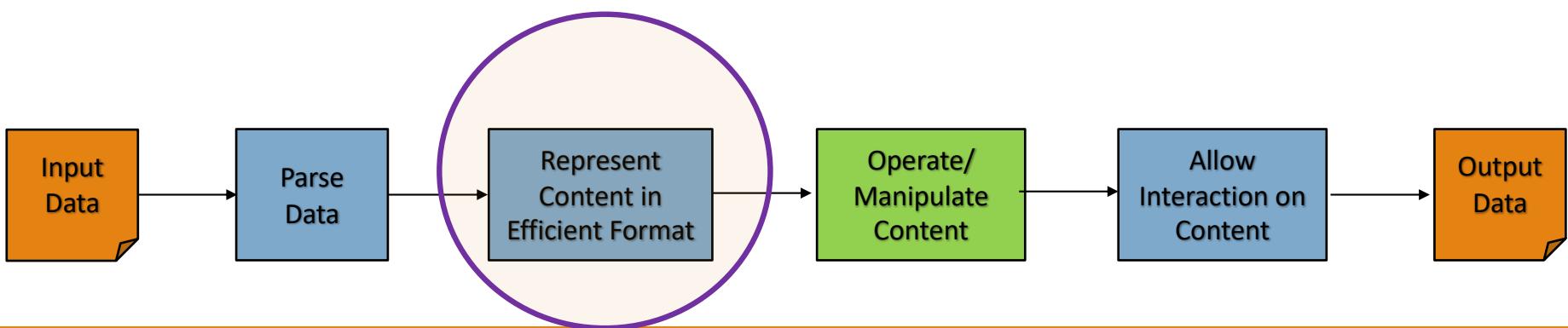


Relationship Between Main AI Topics (Covered in Course)

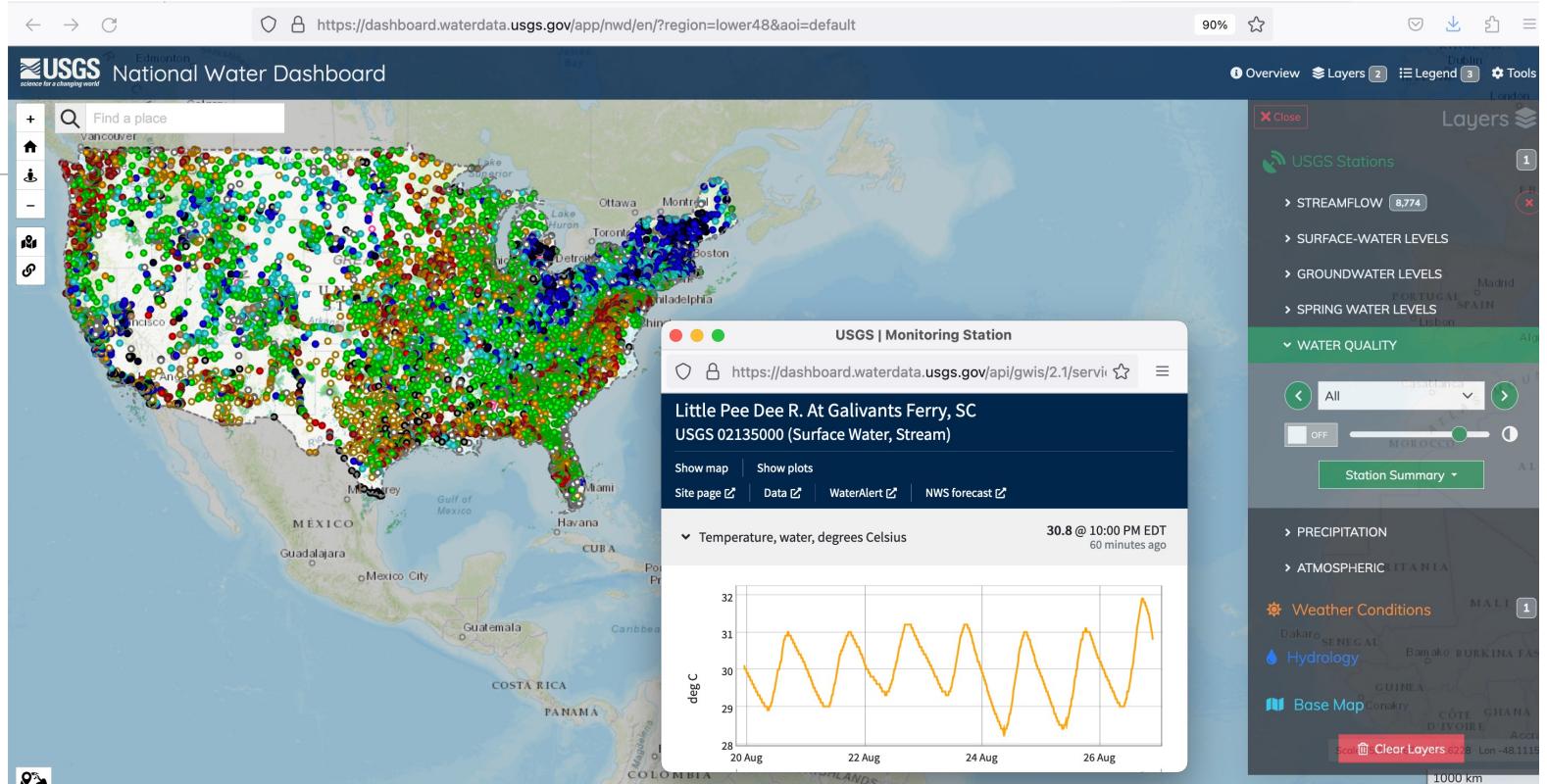


Main Section

Data Processing Pipeline



Water Data



<https://dashboard.waterdata.usgs.gov/app/nwd/en/?region=lower48&aoi=default>

Claims data from 13,000 locations online as on 23 Aug

2025 <https://dashboard.waterdata.usgs.gov/app/nwd/en/?region=lower48&aoi=default>

<https://dashboard.waterdata.usgs.gov/app/nwd/en/?region=lower48&aoi=default>

How Do We Start Working With This?

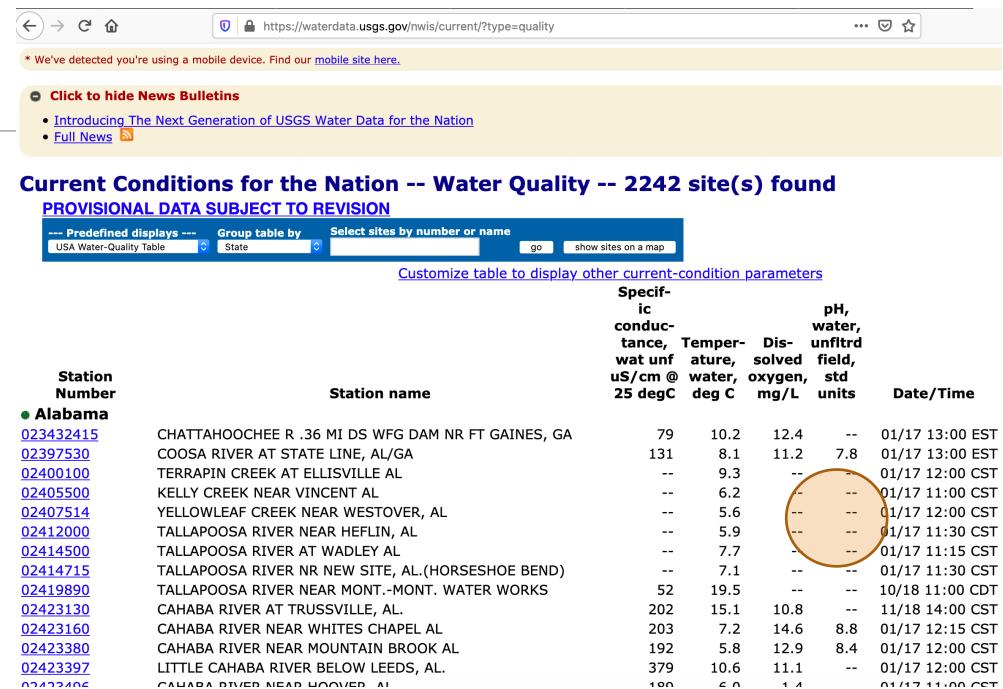
- Access and licensing (Class 2)
- Cleaning, organizing and finding related information (Class 3 – this class)
- Representing formally (in logic) to draw insights (using inferencing) – next week

Is this important ? YES !

- Understanding impact of hurricanes
- Planning during regular times – homes, schools, roads; hospital services; electricity, ...
- Economic development

Common Problem: Missing Value

- Occurrence
 - Missing completely at random
 - Missing at random (a group not wanting to participate)
 - Missing not at random (a group not able to participate)
- What does it mean?
 - The value was not provided
 - The value does not exist or has no practical interpretation
 - The value is being hidden (redaction)
 - Others: The value is not reliable, ...
- How to detect it?
 - By checking for specific values: NA, Not applicable, out-of-range value, 0, -1, "".



The screenshot shows a web browser displaying the USGS Water Data for the Nation website. The URL is https://waterdata.usgs.gov/nwis/current/?type=quality. The page title is "Current Conditions for the Nation -- Water Quality -- 2242 site(s) found". The main content is a table of water quality data for Alabama. The columns include Station Number, Station name, Specific conductance, Temperature, Dissolved oxygen, pH, Date/Time, and various unfiltered parameters. A red circle highlights the "Date/Time" column.

Station Number	Station name	Specific conductance 25 degC	Temperature wat unf deg C	Dissolved oxygen mg/L	pH, water, unfiltrd std units	Date/Time
023432415	CHATTahoochee R .36 MI DS WFG DAM NR FT GAINES, GA	79	10.2	12.4	--	01/17 13:00 EST
02397530	COOSA RIVER AT STATE LINE, AL/GA	131	8.1	11.2	7.8	01/17 13:00 EST
02400100	TERRAPIN CREEK AT ELLISVILLE AL	--	9.3	--	--	01/17 12:00 CST
02405500	KELLY CREEK NEAR VINCENT AL	--	6.2	--	--	01/17 11:00 CST
02407514	YELLOWLEAF CREEK NEAR WESTOVER, AL	--	5.6	--	--	01/17 12:00 CST
02412000	TALLAPOOSA RIVER NEAR HEFLIN, AL	--	5.9	--	--	01/17 11:30 CST
02414500	TALLAPOOSA RIVER AT WADLEY AL	--	7.7	--	--	01/17 11:15 CST
02414715	TALLAPOOSA RIVER NR NEW SITE, AL.(HORSESHOE BEND)	--	7.1	--	--	01/17 11:30 CST
02419890	TALLAPOOSA RIVER NEAR MONT.-MONT. WATER WORKS	52	19.5	--	--	10/18 11:00 CDT
02423130	CAHABA RIVER AT TRUSSVILLE, AL	202	15.1	10.8	--	11/18 14:00 CST
02423160	CAHABA RIVER NEAR WHITES CHAPEL AL	203	7.2	14.6	8.8	01/17 12:15 CST
02423380	CAHABA RIVER NEAR MOUNTAIN BROOK AL	192	5.8	12.9	8.4	01/17 12:00 CST
02423397	LITTLE CAHABA RIVER BELOW LEEDS, AL.	379	10.6	11.1	--	01/17 12:00 CST
02423406	CAHABA RIVER NEAR HOOVER, AL	190	5.0	1.4	--	01/17 11:00 CST

Missing Value – Handling

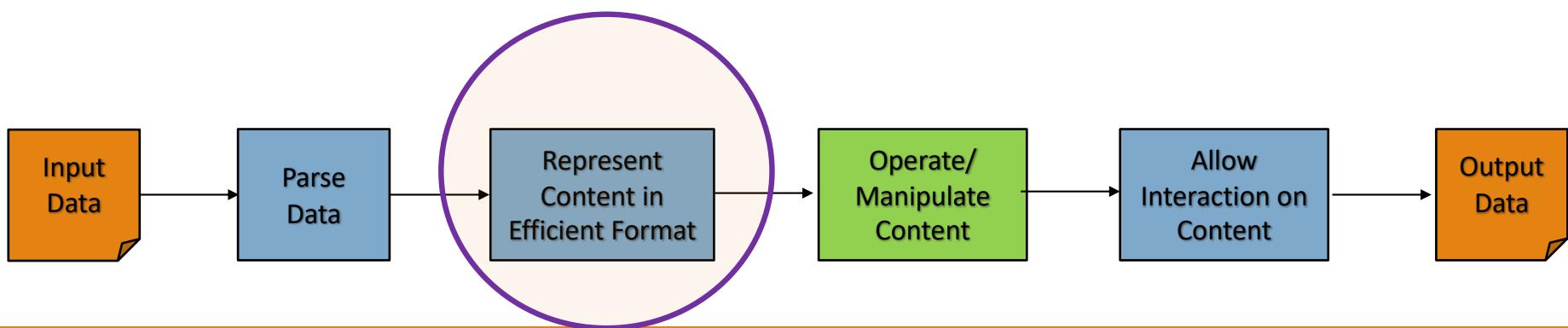
- Ignoring missing value (Omission)
 - Reduces available data
- Impute new value (Imputation)
 - Mean or median
 - Default value
- Analysis techniques which are robust against missing value
 - Expectation maximization

Code Examples

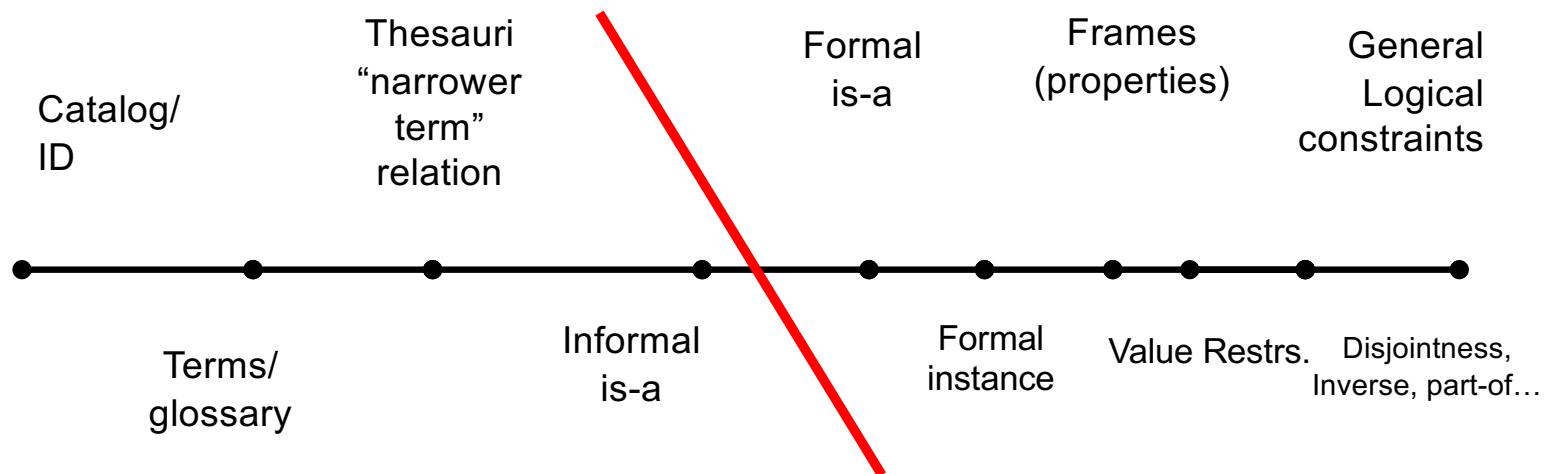
<https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l5-dataprep/>

- Basic concepts: **DataPreparation-Numeric.ipynb**
- An illustration: **Clean-RealSample.ipynb**

Annotation: Knowledge Graphs and Ontology



The Spectrum of Annotation Methods



Ontologies Come of Age McGuinness, 2001, and From AAAI Panel 99 – McGuinness, Welty, Uschold, Gruninger, Lehmann
Plus basis of Ontologies Come of Age – McGuinness, 2003

Thesaurus – Authoritative Entities and Relationships

Countries: https://en.wikipedia.org/wiki/List_of_ISO_3166_country_codes

ISO 3166 ^[1]				ISO 3166-1 ^[2]			ISO 3166-2 ^[3]	
Country name ^[5]	Official state name ^[6]	Sovereignty ^[6] [7][8]	Alpha-2 code ^[5]	Alpha-3 code ^[5]	Numeric code ^[5]	Subdivision code links ^[3]	Internet ccTLD ^[9]	
Afghanistan	The Islamic Republic of Afghanistan	UN member state	AF	AFG	004	ISO 3166-2:AF	.af	
Akrotiri and Dhekelia – See United Kingdom, The								
Åland Islands	Åland	Finland	AX	ALA	248	ISO 3166-2:AX	.ax	
Albania	The Republic of Albania	UN member state	AL	ALB	008	ISO 3166-2:AL	.al	
Algeria	The People's Democratic Republic of Algeria	UN member state	DZ	DZA	012	ISO 3166-2:DZ	.dz	
American Samoa	The Territory of American Samoa	United States	AS	ASM	016	ISO 3166-2:AS	.as	
Andorra	The Principality of Andorra	UN member state	AD	AND	020	ISO 3166-2:AD	.ad	
Angola	The Republic of Angola	UN member state	AO	AGO	024	ISO 3166-2:AO	.ao	
Anguilla	Anguilla	United Kingdom	AI	AIA	660	ISO 3166-2:AI	.ai	
Antarctica ^[a]	All land and ice shelves south of the 60th parallel south	Antarctic Treaty	AQ	ATA	010	ISO 3166-2:AQ	.aq	
Antigua and Barbuda	Antigua and Barbuda	UN member state	AG	ATG	028	ISO 3166-2:AG	.ag	
Argentina	The Argentine Republic	UN member state	AR	ARG	032	ISO 3166-2:AR	.ar	

(Unique) US Counties Information

In COVID sample code: <https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/I3-health/CovidExploration.ipynb>,

reference made to **FIPS** code

References:

- https://www.nrcs.usda.gov/wps/portal/nrcs/detail/national/home/?cid=nrcs143_013697
- https://github.com/kjhealy/fips-codes/blob/master/county_fips_master.csv

Question: how many Richland counties are there in US ?

Answer: 14

County FIPS Codes

FIPS	Name	Stat
01001	Autauga	AL
01003	Baldwin	AL
01005	Barbour	AL
01007	Bibb	AL
01009	Blount	AL
01011	Bullock	AL
01013	Butler	AL
01015	Calhoun	AL
01017	Chambers	AL
01019	Cherokee	AL
01021	Chilton	AL
01023	Choctaw	AL
01025	Clarke	AL
01027	Clay	AL
01029	Cleburne	AL
01031	Coffee	AL
01033	Colbert	AL
01035	Conecuh	AL

Is-a Relationship

List of Countries, States, ... (County), City

- United Nations: <https://unece.org/trade/cefact/unlocode-code-list-country-and-territory>
- US Source: <https://github.com/grammakov/USA-cities-and-states>

Schema.org

- Website: <https://schema.org/docs/about.html>
- GitHub: <https://github.com/schemaorg/schemaorg>
- An organization of metadata information for entities found on the web. Mostly backed by web search companies.
- Explore
 - Thing: <https://schema.org/Thing>
 - Product:

Schema.org

Example 2

No Markup Microdata RDFa JSON-LD Structure

Example notes or example HTML without markup.

```

Dell UltraSharp 30" LCD Monitor

87 out of 100 based on 24 user ratings

$1250 to $1495 from 8 sellers

Sellers:
<a href="save-a-lot-monitors.com/dell-30.html">
  Save A Lot Monitors - $1250</a>
<a href="jondoe-gadgets.com/dell-30.html">
  Jon Doe's Gadgets - $1350</a>
...

```

No structure

Schema.org

Example 2

No Markup Microdata RDFa JSON-LD Structure

Example notes or example HTML without markup.

```

Dell UltraSharp 30" LCD Monitor
87 out of 100 based on 24 user ratings
$1250 to $1495 from 8 sellers
Sellers:
<a href="save-a-lot-monitors.com/dell-30.html">
Save A Lot Monitors - $1250</a>
<a href="jondoe-gadgets.com/dell-30.html">
Jon Doe's Gadgets - $1350</a>
...

```

No structure

Structure in JSON-LD format

Example 2

No Markup Microdata RDFa JSON-LD Structure

Example encoded as JSON-LD in a HTML script tag.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "aggregateRating": {
    "@type": "AggregateRating",
    "bestRating": "100",
    "ratingCount": "24",
    "ratingValue": "87"
  },
  "image": "dell-30in-lcd.jpg",
  "name": "Dell UltraSharp 30\" LCD Monitor",
  "offers": {
    "@type": "AggregateOffer",
    "highPrice": "$1495",
    "lowPrice": "$1250",
    "offerCount": "8",
    "offers": [
      {
        "@type": "Offer",
        "url": "save-a-lot-monitors.com/dell-30.html"
      },
      {
        "@type": "Offer",
        "url": "jondoe-gadgets.com/dell-30.html"
      }
    ]
  }
}</script>
```

Schema.org

Example 2

No Markup Microdata RDFa JSON-LD Structure

Example notes or example HTML without markup.

```

Dell UltraSharp 30" LCD Monitor

87 out of 100 based on 24 user ratings
$1250 to $1495 from 8 sellers

Sellers:
<a href="save-a-lot-monitors.com/dell-30.html">
  Save A Lot Monitors - $1250</a>
<a href="jondoe-gadgets.com/dell-30.html">
  Jon Doe's Gadgets - $1350</a>
...

```

No structure

Example 2

No Markup Microdata RDFa JSON-LD Structure

Example encoded as JSON-LD in a HTML script tag.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Product",
  "aggregateRating": {
    "@type": "AggregateRating",
    "bestRating": "100",
    "ratingCount": "24",
    "ratingValue": "87"
  },
  "image": "dell-30in-lcd.jpg",
  "name": "Dell UltraSharp 30\" LCD Monitor",
  "offers": [
    {
      "@type": "AggregateOffer",
      "highPrice": "$1495",
      "lowPrice": "$1250",
      "offerCount": "8",
      "offers": [
        {
          "@type": "Offer",
          "url": "save-a-lot-monitors.com/dell-30.html"
        },
        {
          "@type": "Offer",
          "url": "jondoe-gadgets.com/dell-30.html"
        }
      ]
    }
  ]
}</script>
```

Structure in JSON-LD format

Example 2

No Markup Microdata RDFa JSON-LD Structure

Structured representation of the JSON-LD example.

@type	Product
name	Dell UltraSharp 30" LCD Monitor
offers	
@type	AggregateOffer
offerCount	8
lowPrice	\$1250
highPrice	\$1495
offers	
@type	Offer
url	http://example.org/jondoe-gadgets.com/dell-30.html
offers	
@type	Offer
url	http://example.org/save-a-lot-monitors.com/dell-30.html
image	http://example.org/dell-30in-lcd.jpg
aggregateRating	
@type	AggregateRating
ratingValue	87
ratingCount	24
bestRating	100

Induced Structure

Schema.org - continued

- **Exploration Exercise**

- Services: <https://schema.org/Service>
- Event: <https://schema.org/Event>

- Benefit:

- Easy to incorporate annotations
- Uses popular development tools and technologies (JSON, Microformat)

- Disadvantage

- Cannot perform deep inferencing
- Popular in certain communities

Formalizing Knowledge in an Ontology

Sources:

Achille Fokoue, Anastasios Kementsietsidis Tutorial

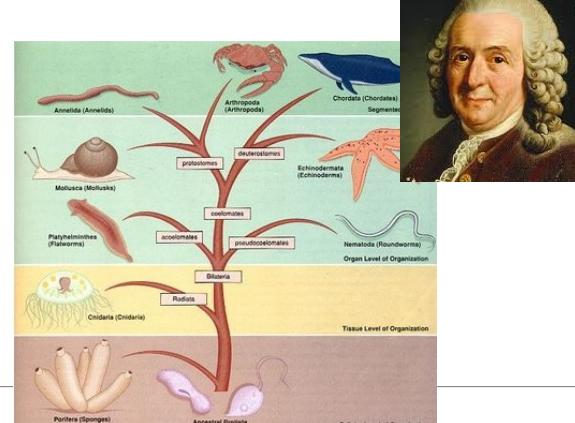
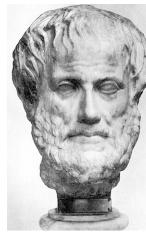
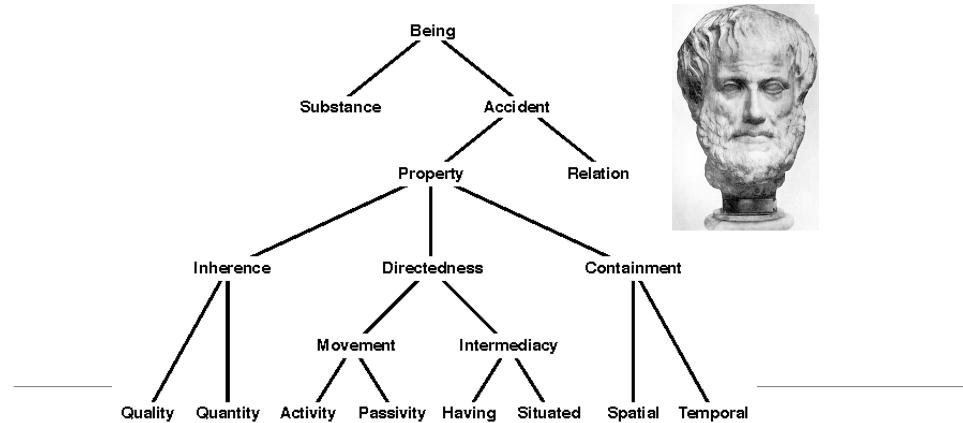
SCRIBE presentation by Rosario Usceda Sosa, Biplav Srivastava, Bob Schloss

- <https://github.com/rschloss/ismp>,
- https://researcher.watson.ibm.com/researcher/view_group.php?id=2505

What is an ontology, anyway?

In Computer Science, “An ontology is a formal explicit description of concepts in a domain of discourse (**classes** (sometimes called concepts)), **properties** of each concept describing various features and **attributes** of the concept (slots (sometimes called roles or properties)), and **restrictions** on slots (facets (sometimes called role restrictions)). An ontology together with a set of individual instances of classes constitutes a knowledge base. In reality, there is a fine line where the ontology ends and the knowledge base begins.” [Noy, 2000]

Not to be confused with ontologies (and/or taxonomies) in Philosophy or Life Sciences

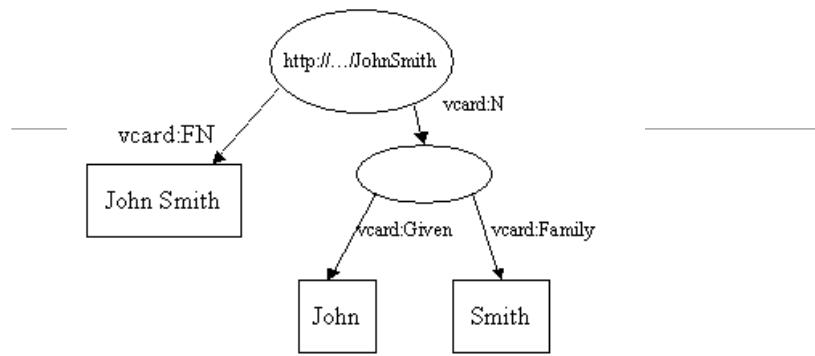


In a Smart City domain, we’re concerned with modeling the *city data* (city activity data, city departments, assets, KPIs), not the city itself (the full set of spatial and temporal relations between people and objects in the city) Ontologies help us to structure and reason about city events, entities and services.

Ontology = Class + Relations + Constraints

Knowledge Base = Ontology + instances + (Standard) Inference and rules

RDF / Turtle Example



---- Turtle ----

```
<http://somewhere/JohnSmith>
  <http://www.w3.org/2001/vcard-rdf/3.0#FN>
    "John Smith" ;
  <http://www.w3.org/2001/vcard-rdf/3.0#N>
    [ <http://www.w3.org/2001/vcard-
      rdf/3.0#Family>
        "Smith" ;
      <http://www.w3.org/2001/vcard-
      rdf/3.0#Given>
        "John"
    ] .
```

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-
rdf/3.0#" >
  <rdf:Description rdf:nodeID="A0">
    <vcard:Given>John</vcard:Given>
    <vcard:Family>Smith</vcard:Family>
  </rdf:Description>
  <rdf:Description
    rdf:about="http://somewhere/JohnSmith">
    <vcard:FN>John Smith</vcard:FN>
    <vcard:N rdf:nodeID="A0"/>
  </rdf:Description>
</rdf:RDF>
```

OWL extends RDF...

RDF-schema

- Class, subclass
- Property, subproperty

+ Restrictions

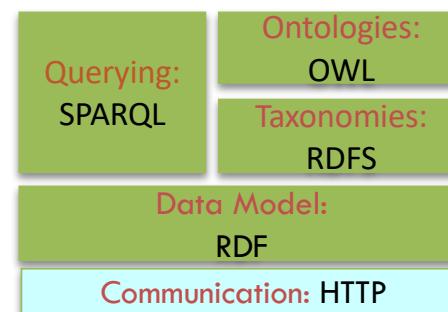
- Range, domain
- Local, global
- Existential
- Cardinality

+ Combinators

- Union, Intersection
- Complement
- Symmetric, transitive

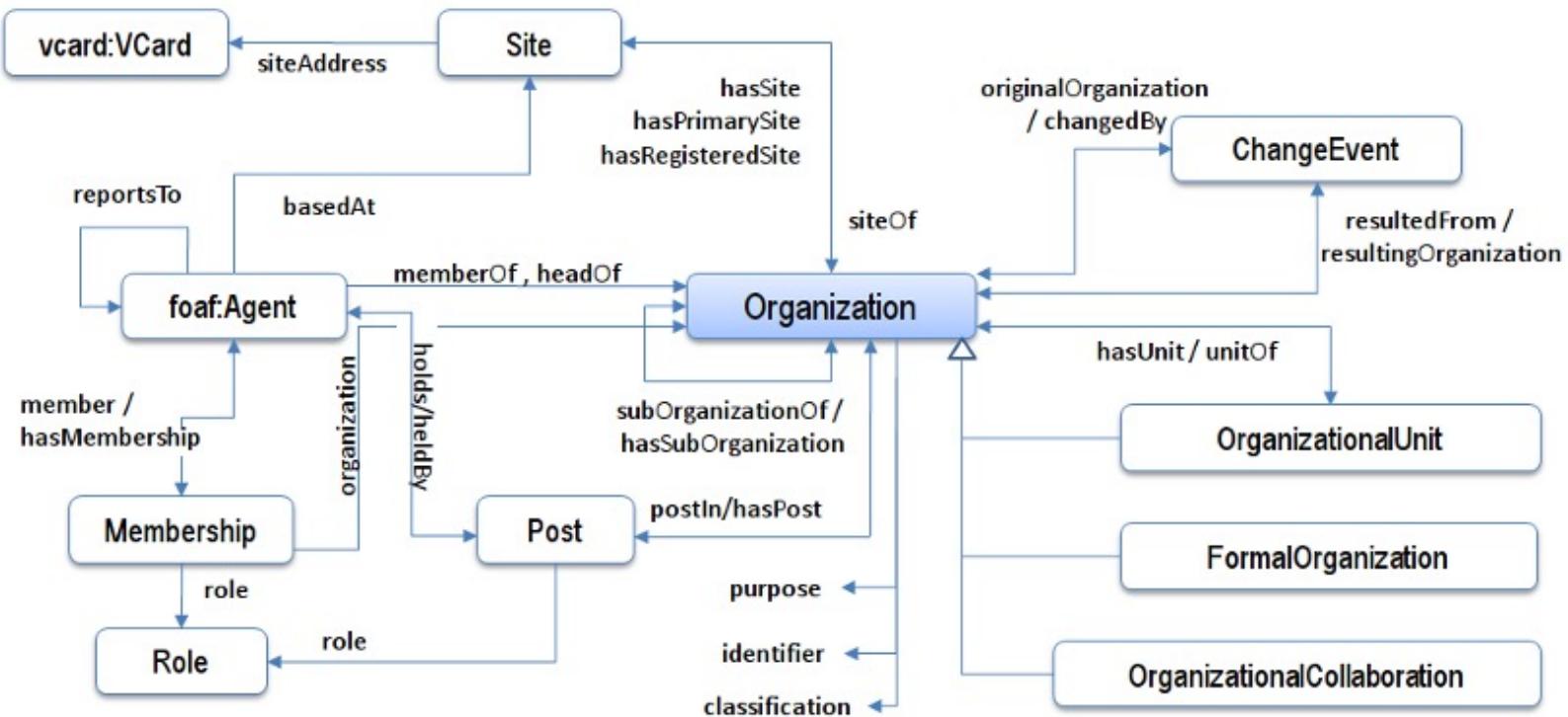
+ Mapping

- Equivalence
- Inverse



Source: Achille Fokoue, Anastasios Kementsietsidis Tutorial

Larger Example: Organization Ontology



Ontology description: <http://www.w3.org/TR/vocab-org/>

Ontology: <http://www.w3.org/ns/org.ttl>

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

...
@prefix : <http://www.w3.org/ns/org#> .

# -- Meta data ----

<http://www.w3.org/ns/org#>
  a owl:Ontology;
  owl:versionInfo "0.7";
  rdfs:label "Core organization ontology"@en;
  rdfs:comment "Vocabulary for describing organizational structures, specializable to a broad variety of types of organization."@en;
  dct:created "2010-05-28"^^xsd:date;
  dct:modified "2010-06-09"^^xsd:date;
  dct:modified "2010-10-08"^^xsd:date;
...
  rdfs:seeAlso <http://www.w3.org/TR/vocab-org/> ;
.

# -- Organizational structure ----
-
org:Organization a owl:Class, rdfs:Class;
  rdfs:subClassOf foaf:Agent;
  owl:equivalentClass foaf:Organization;
  rdfs:label "Organization"@en;
  rdfs:label "Organisation"@fr;
  owl:hasKey (org:identifier) ;
  rdfs:comment """Represents a collection of people organized together into a community or other social, commercial or political structure. ... Alternative names: _Collective_ _Body_ _Org_ _Group_ """@en;
  rdfs:comment """Représente un groupe de personnes organisées en communauté où tout autre forme de structure sociale, commerciale ou politique. ... code provenant d'une liste de code."""@fr;
  rdfs:isDefinedBy <http://www.w3.org/ns/org> ;
.

```

<http://www.w3.org/ns/org.ttl>

Larger Ontology

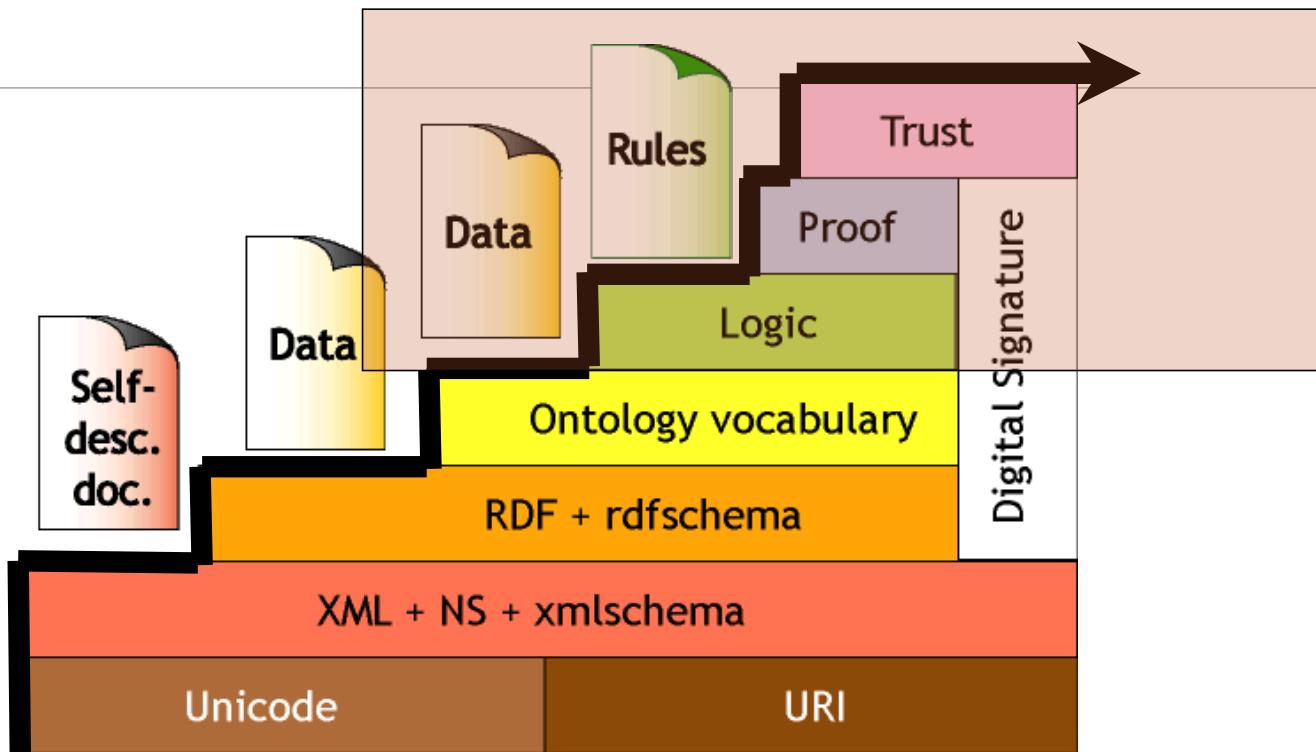
```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#" xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:org="http://www.w3.org/ns/org#" xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:dct="http://purl.org/dc/terms/"
  xmlns:prov="http://www.w3.org/ns/prov#" xmlns:owlTime="http://www.w3.org/2006/time#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:vcard="http://www.w3.org/2006/vcard/ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
+ <owl:Ontology rdf:about="http://www.w3.org/ns/org#">
+ <rdfs:Class rdf:about="http://www.w3.org/ns/org#Organization">
- <rdfs:Class rdf:about="http://www.w3.org/ns/org#Role">
  <rdfs:label xml:lang="fr">Rôle</rdfs:label>
- <owl:disjointWith>
  <owl:Class rdf:about="http://www.w3.org/ns/org#ChangeEvent" />
</owl:disjointWith>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2004/02/skos/core#Concept" />
- <owl:disjointWith>
  <owl:Class rdf:about="http://www.w3.org/ns/org#Site" />
</owl:disjointWith>
<rdfs:comment xml:lang="fr">Indique le rôle qu'une Personne ou un autre Agent peut avoir dans une Organisation. Les instances de cette classe décrivent le rôle dans l'absolu; pour indiquer une personne ayant ce rôle spécifique dans une Organisation, utilisez une instance de `org:Membership`. Il est courant que les rôles soient organisés dans une sorte de taxonomie, ce qui peut être représenté avec SKOS. Les propriétés de libellés standards de SKOS devraient être utilisées pour libeller le Rôle. D'autres propriétés additionnelles pour ce rôle, comme une fourchette de Salaire peuvent être ajoutées par une extension de ce vocabulaire.</rdfs:comment>
- <owl:disjointWith>
  <owl:Class rdf:about="http://www.w3.org/ns/org#Membership" />
</owl:disjointWith>
<rdfs:label xml:lang="en">Role</rdfs:label>
<rdfs:isDefinedBy rdf:resource="http://www.w3.org/ns/org" />
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class" />
<rdfs:comment xml:lang="en">Denotes a role that a Person or other Agent can take in an organization. Instances of this class describe the abstract role; to denote a specific instance of a person playing that role in a specific organization use an instance of `org:Membership` It is common for roles to be

```

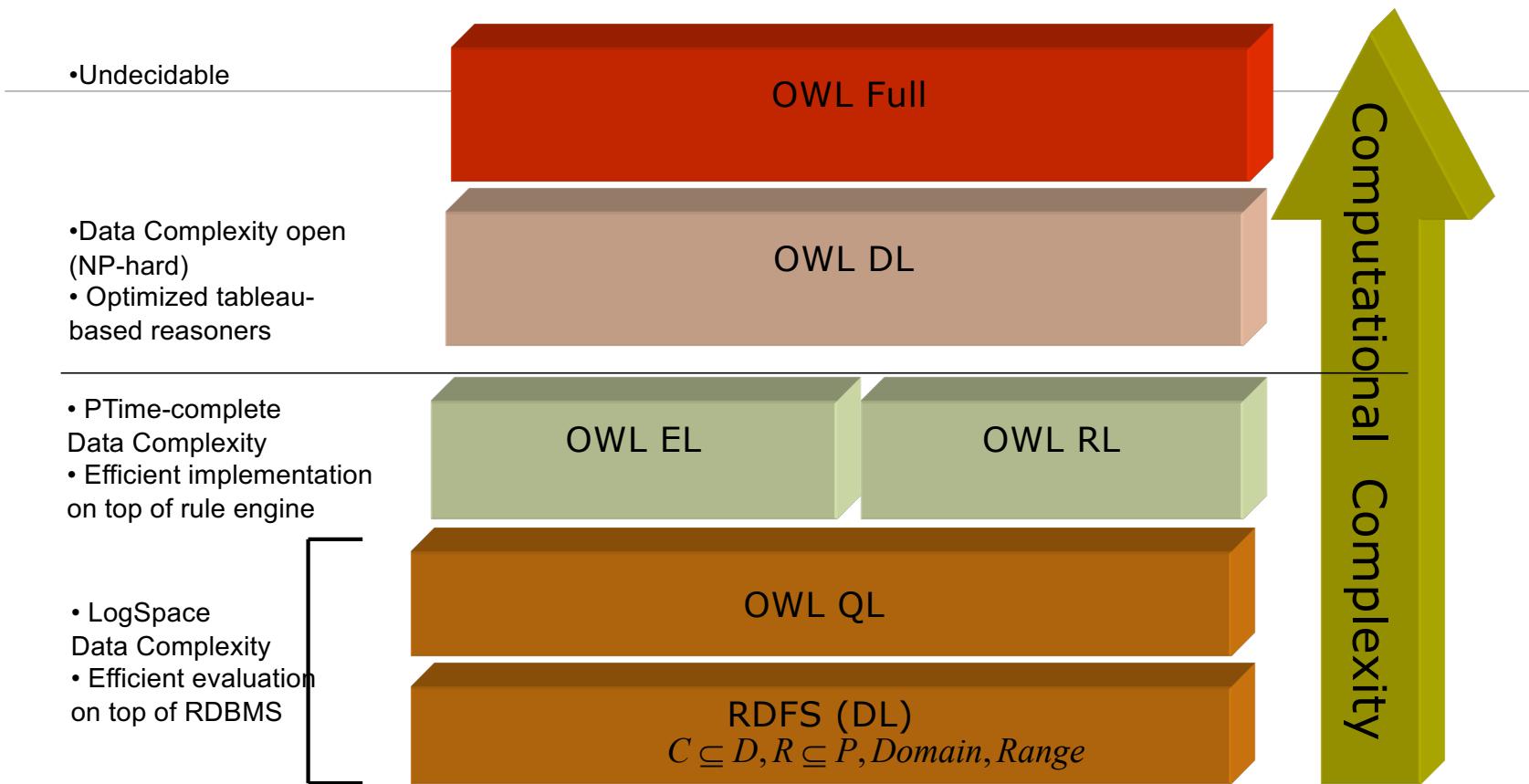
<http://www.w3.org/ns/org>

Moving to the future of the web



Semantic Web LayerCake (Berners-Lee, 99; Swartz-Hendler, 2001)

Challenge of Reasoning on Ontologies



Not all ontologies are created equal

In practice, ontologies are used -together with inferencing engines and rules-, for a variety of purposes. If we think of them as schemas, there are different ways

	Purpose	Instances	Inferencing	Examples
Normative schema	As a deductive system	Deductive System (axioms + deductive rules)	Part of the knowledge base	Defined by rules. Expert systems, Planning, Optimization.
	As a data blueprint	Constrain a domain	Must conform to the normative schema determined by the ontology	Subsumption, class inferencing Biomedical and life sciences (FMA, Radlex)
	As a data classifier	Classify open data	Unknown formats	Subsumption, class inferencing Tag ontologies (MOAT, Echarte, SCOT, NAO, etc.)
	As a data integrator	Integrating pre-defined model to existing data sources	Instances are mapped, no constraint enforcement.	Subsumption, class, entity inferencing SCRIBE
	As data mapping vocabulary	Mapping to/from existing data sources	Mined instances determine the ontology/schema.	Subsumption, class inferencing D2RQ (a tool)
Integrative Schema, depend on instances				

SCRIBE belongs to the **fourth** category: It has no constraints and was designed to support the programming of tools that allow domain experts to deal with entities natural to them (even if the recorded data is actually distributed).

What makes a good ontology for data integration?

A *good* ontology is a *useful* ontology, an ontology that *both* humans and systems can process.

Human Usability

Communicable. Naming, natural language support, etc.

Concise. A simple way to describe the key entities of the model and yet able to infer many facts

Consistent. Naming conventions and modeling patterns

Authoritative to domain experts

Documented, not just descriptions, but also provenance

Managed and maintained by people throughout the model lifecycle.

Reusable in similar domains, for similar instances.

- ❑ *Formal representation of knowledge in a particular domain*
- ❑ *Formally defines key concepts and relations in the domain*
- ❑ *Specifies relationships between those key concepts and relations*
- ❑ *Supports automated reasoning about entities in the domain*

System Usability

Scalable so large amounts of data can be parsed, stored and retrieved.

Efficient query and inferencing

Programmable solutions, both in open and closed data paradigms.

Open infrastructure and tools

Using Ontology

- Visually via tools like Protégé - <https://protege.stanford.edu/>
- Programmatically with APIs like
 - Jena (Java) - <https://jena.apache.org/documentation/ontology/>
 - OwlReady2 (Python) - <https://bitbucket.org/jibalamy/owlready2/src/master/>
 - Rdflib (Python) - <https://github.com/RDFLib/OWL-RL>
- A compendium of resources - <https://github.com/totogo/awesome-knowledge-graph>

Code Illustration

On Github:

<https://github.com/biplav-s/course-nl/blob/master/l11-ontology/Exploring%20ontologies.ipynb>

Knowledge Graph

- No clear definition
 - ["Towards a Definition of Knowledge Graphs," by Lisa Eherlinger and Wolfram Wöß, CEURWorkshop Proceedings. 2016, <http://ceur-ws.org/Vol-1695/paper4.pdf>](http://ceur-ws.org/Vol-1695/paper4.pdf)
 - For practical purposes, concepts and their relationships; not constraints
 - Driven by applications in search and information integration
 - See discussion at: <http://accidental-taxonomist.blogspot.com/2019/05/knowledge-graphs-and-ontologies.html>
- But ontology as knowledge graph widely used in industries
 - Industry-Scale Knowledge Graphs: Lessons and Challenges, CACM 2019, <https://cacm.acm.org/magazines/2019/8/238342-industry-scale-knowledge-graphs/fulltext>

KG Usage

	Data model	Size of the graph	Development stage
Microsoft	The types of entities, relations, and attributes in the graph are defined in an ontology.	~2 billion primary entities, ~55 billion facts	Actively used in products
Google	Strongly typed entities, relations with domain and range inference	1 billion entities, 70 billion assertions	Actively used in products
Facebook	All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal.	~50 million primary entities, ~500 million assertions	Actively used in products
eBay	Entities and relation, well-structured and strongly typed	Expect around 100 million products, >1 billion triples	Early stages of development and deployment
IBM	Entities and relations with evidence information associated with them.	Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million	Actively used in products and by clients

Figure courtesy: Industry-Scale Knowledge Graphs: Lessons and Challenges, CACM 2019

Lecture 3: Summary

- We talked about
 - Data preparation
 - Knowledge representation/ graph
 - Ontology

Projects A: Start (4 weeks; 200 points)

- End date: **Thursday, Sep 18**
- Model AI Assignments: <http://modelai.gettysburg.edu/>
 - Choose a project, preferably within last 5 years (i.e., after 2020).
 - Enter its name in “Student-InfoShared ..” sheet, column G
 - Follow instructions and do it alone
 - Submit project outcome
 - Create a project on your Github called **ProjectA**.
 - Put deliverables, as per project description, inside the folder, and commit.
 - Timestamp will be used to confirm that Project-A is done on time

Lecture 4: Data and Trust Issues

- Start Machine Learning
- Attention to
 - Data issues
 - Trust issues
 - Evaluation

Main Section



Credit: Retrieved from internet

Machine Learning – Insights from Data

- Descriptive analysis
 - Describe a past phenomenon
 - **Methods:** classification (feedback from label), clustering, dimensionality reduction, anomaly detection, neural methods, reinforcement learning (feedback from hint/ reward)
- Predictive analysis
 - Predict about a new situation
 - **Methods:** time-series, neural networks
- Prescriptive analysis
 - What an agent should do
 - **Methods:** simulation, reinforcement learning, reasoning
- New areas
 - Counterfactual analysis
 - Causal Inferencing
 - Scenario planning

Nomenclature

Column, Attribute, Feature

Row, Item

1	PID	ST_NUM	ST_NAME	OWN_OCCUPIED	NUM_BEDROOMS	NUM_BATH	SQ_FT
2	100001000	104	PUTNAM	Y	3	1	1000
3	100002000	197	LEXINGTON	N	3	1.5	--
4	100003000		LEXINGTON	N	n/a	1	850
5	100004000	201	BERKELEY	12	1	NaN	700
6		203	BERKELEY	Y	3	2	1600
7	100006000	207	BERKELEY	Y	NA	1	800
8	100007000	NA	WASHINGTON		2	HURLEY	950
9	100008000	213	TREMONT	Y	1	1	
10	100009000	215	TREMONT	Y	na	2	1800

Types of Attributes/ Columns

- Numeric: has number as value in computational sense; all mathematical functions are valid.
 - Example: SQ_FT
- Categorical: has distinct values
 - Nominal: each value is incomparable with other
 - Example: OWN_OCCUPIED, ST_NAME
 - Ordinal: the values can be ordered
 - Example: ST_NUM, NUM_BEDS
- Comment:
 - Q: what type is a binary variable?
 - A: depends on the semantics – nominal (gender), ordinal (number basements).

	PID	ST_NUM	ST_NAME	OWN_OCCUPIED	NUM_BEDROOMS	NUM_BATH	SQ_FT
1	100001000	104	PUTNAM	Y	3	1	1000
2	100002000	197	LEXINGTON	N	3	1.5	--
3	100003000		LEXINGTON	N	n/a	1	850
4	100004000	201	BERKELEY	12	1	NaN	700
5		203	BERKELEY	Y	3	2	1600
6	100006000	207	BERKELEY	Y	NA	1	800
7	100007000	NA	WASHINGTON		2	HURLEY	950
8	100008000	213	TREMONT	Y	1	1	
9	100009000	215	TREMONT	Y	na	2	1800

Why is Type of Variable Important

- Handling of missing values
- Distance between
 - Values
 - Data items
- Used for measuring accuracy, error
- Guiding the learning process
 - Selection of algorithms

Concepts

- **Input data:** data available
 - **Training data:** used for training a learning algorithm and get a model
 - [Optional] **Validation data:** used to tune parameters
- **Test data:** used to test a learning model

- **Classification problem**

- Separating data into classes (also called labels, categorical types)
- One of the attributes is the class label we are trying to learn
- Class label is the **supervision**

- **Clustering problem**

- We are trying to learn grouping of data
- There is no attribute indicating membership in the groups (hence, **unsupervised**)

- **Prediction problem**

- Learning value of a continuous variable

Reference: <https://machinelearningmastery.com/difference-test-validation-datasets/>
<https://www2.seas.gwu.edu/~bell/csci243/lectures/classification.pdf>

Discussion: Learning Task

- Task: For living beings, predicting their weights and heights
- Task-h: Model for humans
 - Setting-1: Features: name, weight; output height
 - Setting-2: Features: name, height; output weight
- Task-a: Model for animal
 - Task: a_d: for dogs
 - Task: a_c: for cats
 - ...

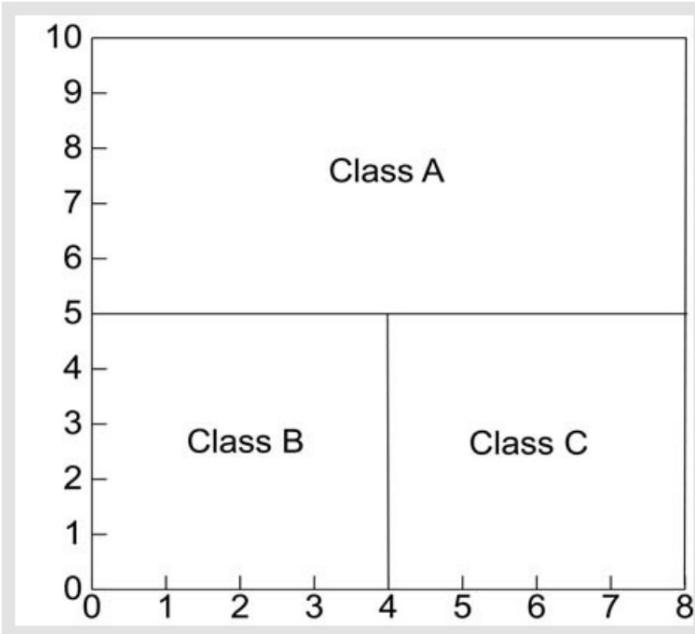
Sample Learning Task

- COVID-19 data

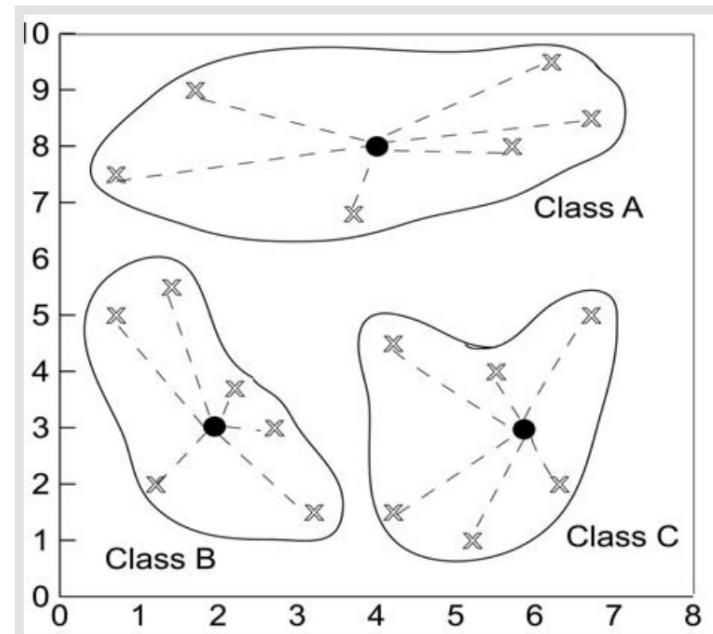
Notebook: <https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l6-l7-l8-supervised-ml/Supervised-Regression-Classification.ipynb>

Methods for Classification

Partitioning Based



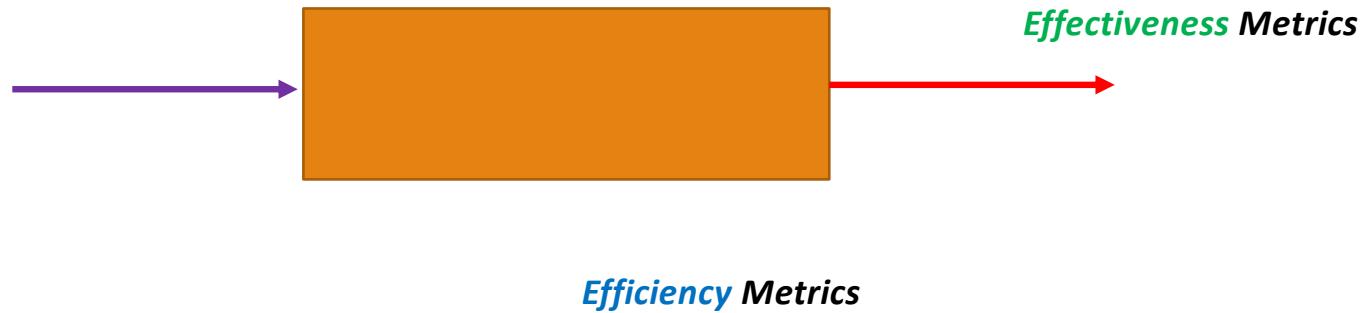
Distance Based



Source: <https://www2.seas.gwu.edu/~bell/csci243/lectures/classification.pdf>

Metric Types

- **Effectiveness**: what the user of a system sees, primarily cares about
- **Efficiency**: what the executor in a system sees, primarily cares about



Example: Predicting COVID cases

- **Effectiveness:** what the user of a system sees, primarily cares about
 - *How accurate (high) is the prediction?*
 - *How low is the error?*
- **Efficiency:** what the executor in a system sees, primarily cares about
 - *How low is the error?*
 - *How fast was prediction made?*
 - *How stable is the prediction to change in data?*

Example: Detecting Spam in Email

- **Effectiveness:** what the user of a system sees, primarily cares about
 - *How many spams identified?*
 - *How many spams missed?*
- **Efficiency:** what the executor in a system sees, primarily cares about
 - *How fast were spams detected?*
 - *How much memory was used per million emails processed ?*

Comparing Classification Methods

- Predictive accuracy
 - Interpretability: providing insight
 - Robustness: handling noisy data
-
- Speed
 - Scalability: large volume of data

Source: Data Mining: Concepts and Techniques, by Jiawei Han and Micheline Kamber

Metrics: Accuracy, Precision, Recall

		Predicted class	
Actual Class		Class = Yes	Class = No
		Class = Yes	True Positive
	Class = No	False Positive	True Negative

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)}$$

$$\text{Precision} = \frac{(TP)}{(TP+FP)}$$

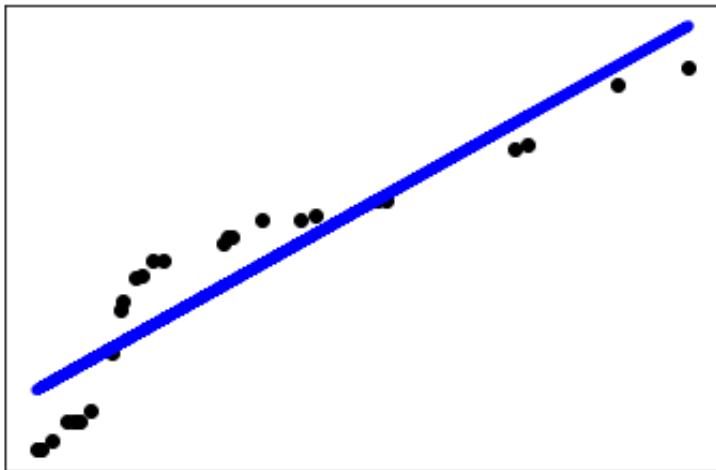
$$\text{Recall} = \frac{(TP)}{(TP+FN)}$$

F1 Score: Harmonic Mean

$$1/F1 = 1/\text{Precision} + 1/\text{Recall}$$

$$F1 = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

Linear Regression



Notebook: <https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l6-l7-l8-supervised-ml/Supervised-Regression.ipynb>

Reference and Demo

- Data: UCI Datasets

- <https://archive.ics.uci.edu/datasets>
- Browse or search

The screenshot shows the homepage of the Weka 3 website. At the top, there is a navigation bar with links for Project, Software, Book, Courses, Publications, People, and Related. Below the navigation bar, the title "Weka 3: Machine Learning Software in Java" is displayed. A brief introduction states: "Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization." A note below says: "Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like this, and the bird sounds like this." Another note mentions: "Weka is open source software issued under the GNU General Public License." It also links to free online courses and mentions deep learning support.

The screenshot shows the "Browse Datasets" page of the UCI Machine Learning Repository. The URL is https://archive.ics.uci.edu/datasets. On the left, there are filters for Keywords, Data Type (Multivariate, Sequential, Tabular, Text, Time-Series, Other), Subject Area, and Task. The main area is titled "Browse Datasets" and shows a list of datasets. The first dataset listed is "Iris", described as a small classic dataset from Fisher, 1936. It is a Classification task using Tabular data with 150 instances and 4 features. The second dataset is "Heart Disease", a Classification task using Multivariate data with 303 instances and 13 features. The third dataset is "Adult", a Classification task using Multivariate data with 48,84K instances and 14 features. The fourth dataset is "Wine", a Classification task using Tabular data with 178 instances and 13 features.

- Tools:

- Weka - <https://www.cs.waikato.ac.nz/ml/weka/>
- Download tool and dataset

- Libraries

- Scikit - <https://scikit-learn.org/stable/>

Reference and Demo

- Data: UCI Datasets
 - <https://archive.ics.uci.edu/datasets>
 - Browse or search
- Tools:
 - Weka -
<https://www.cs.waikato.ac.nz/ml/weka/>
 - Download tool and dataset
- Libraries
 - Scikit - <https://scikit-learn.org/stable/>

The screenshot shows the official scikit-learn website at <https://scikit-learn.org/stable/>. The header includes the scikit-learn logo, navigation links for Install, User Guide, API, Examples, Community, and More, and a search bar.

The main content area features three large cards:

- Classification**: Describes identifying which category an object belongs to. It lists Applications: Spam detection, image recognition and Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more... Below is a grid of 9x3 plots showing various classification results.
- Regression**: Describes predicting a continuous-valued attribute associated with an object. It lists Applications: Drug response, Stock prices and Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more... Below is a line plot titled "Boosted Decision Tree Regression" showing target values versus index, comparing training samples (blue dots) and estimators (orange line).
- Clustering**: Describes automatic grouping of similar objects into sets. It lists Applications: Customer segmentation, Grouping experimental outcomes and Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more... Below is a scatter plot of digits dataset points colored by cluster assignment, with centroids marked by white crosses.

Exercise: German Credit

- Check in UCI
- Look at variants
 - <https://archive.ics.uci.edu/dataset/573/south+german+credit+update>

Concluding Section

Projects A: Start (4 weeks; 200 points)

- End date: **Thursday, Sep 18**
- Model AI Assignments: <http://modelai.gettysburg.edu/>
 - Choose a project, preferably within last 5 years (i.e., after 2020).
 - Enter its name in “Student-InfoShared ..” sheet, column G
 - Follow instructions and do it alone
 - Submit project outcome
 - Create a project on your Github called **ProjectA**.
 - Put deliverables, as per project description, inside the folder, and commit.
 - Timestamp will be used to confirm that Project-A is done on time

Week 2: Concluding Comments

- We looked at
 - L3: Data Prep, Knowledge Graph
 - L4: ML Basics
- Discussion on Projects A

About Week 3 – Lectures 5, 6

Lecture 5, 6: AI / ML Methods

- Supervised ML

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2: Data: Formats, Representation, ML Basics
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots