AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

*CSCE 580: Introduction to AI*

# Lecture 3: Representing and Organizing Data

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

27TH AUG 2024

**Carolinian Creed: "I will practice personal and academic integrity."**

**Credits**: **Copyrights of all material reused acknowledged**

# Organization of Lecture 3

- Introduction Segment
  - Recap of Lecture 2
- Main Segment
  - Data preparation
  - Knowledge representation/ graph
  - Ontology
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 4
  - Ask me anything

# Introduction Section

# Recap of Lecture 2

- Data formats
- Big data v/s open data
- Open data
  - City data
  - Data access via Open311
  - Publishing data systematically

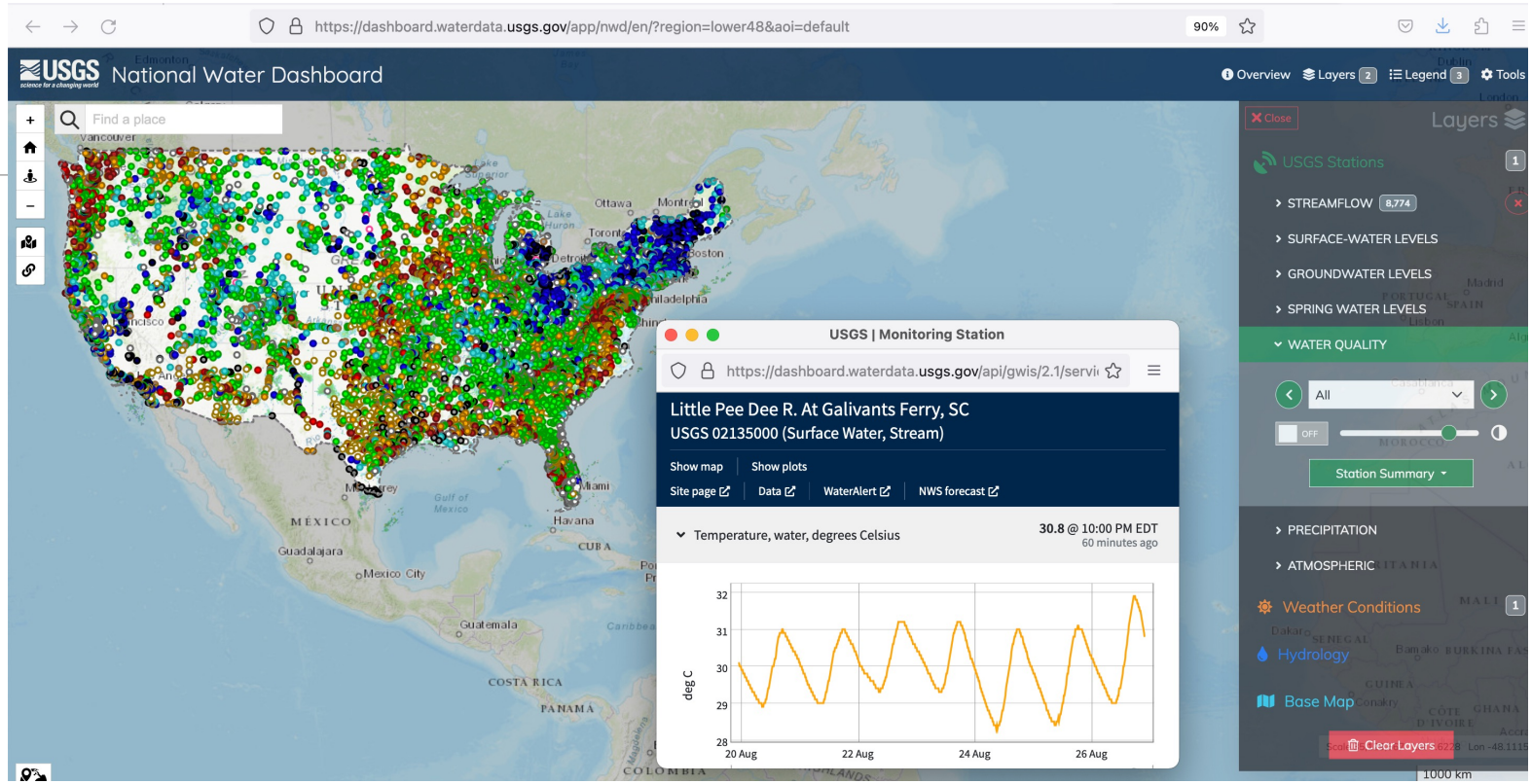
# Where We Are

**CSCE 580/ 581 – In This Course**

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week 8: Machine Learning - Classification
- Week 9: Machine Learning - Classification – Trust Issues and Mitigation Methods
- Topic 10: Learning neural network, deep learning, Adversarial attacks
- Week 11: Large Language Models – Representation, Issues
- Topic 12: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 13: Planning, Reinforcement Learning – Sequential decision making
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

# Main Section

# Water Data




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

Claims data from 13,000 locations online on 26 Aug 2023

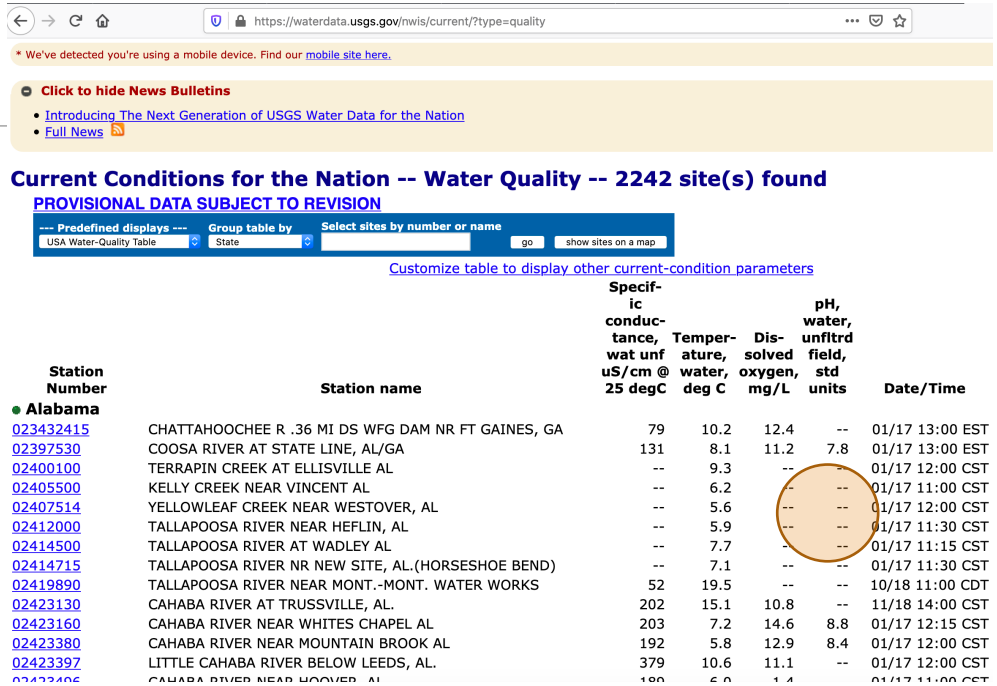
# How Do We Start Working With This?

- 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, “”.



https://waterdata.usgs.gov/nwis/current/?type=quality

* We've detected you're using a mobile device. Find our mobile site here.

Click to hide News Bulletins

- Introducing The Next Generation of USGS Water Data for the Nation
- Full News

**Current Conditions for the Nation -- Water Quality -- 2242 site(s) found**

PROVISIONAL DATA SUBJECT TO REVISION

--- Predefined displays --- | USA Water-Quality Table | Group table by | State | Select sites by number or name | go | show sites on a map

Customize table to display other current-condition parameters

| Station Number | Station name | Specif-ic conduc-tance, wat unf uS/cm @ 25 degC | Temper-ature, water, deg C | Dis-solved oxygen, mg/L | pH, water, unfltrd field, std units | Date/Time |
|---|---|---|---|---|---|---|
| **Alabama** | | | | | | |
| 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 | 189 | 6.9 | 1.4 | | 01/17 11:00 CST |

# Missing Value – Handling

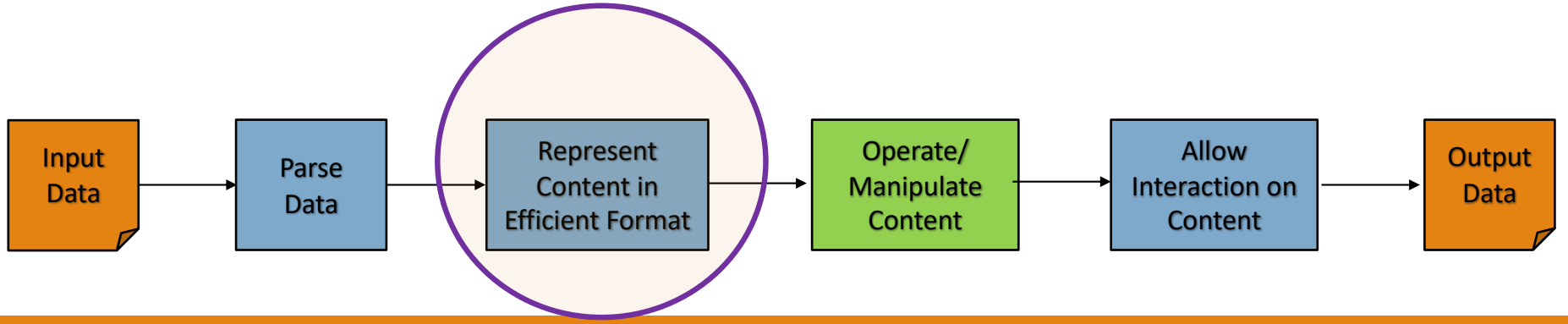
- Ignoring missing value (Omission)
  - Reduces available data
- Impute new value (Imputation)
  - Mean or median
  - Default value
- Analysis techniques which are robust against missing value
  - Expectation maximization

# Code Examples

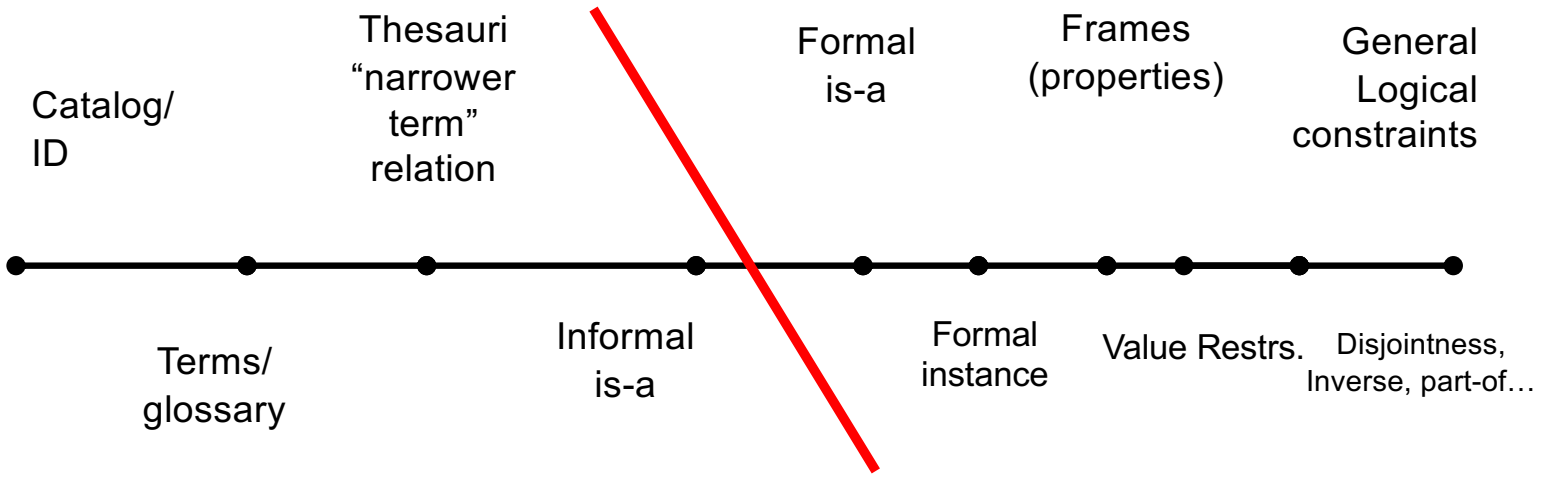
https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l5-dataprep/

- 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/l3-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.*

```
<img src="dell-30in-lcd.jpg" alt="A Dell UltraSharp monitor" />
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.*

```
<img src="dell-30in-lcd.jpg" alt="A Dell UltraSharp monitor" />
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

# Schema.org

Example 2

No Markup | Microdata | RDFa | JSON-LD | Structure

*Example notes or example HTML without markup.*

```
<img src="dell-30in-lcd.jpg" alt="A Dell UltraSharp monitor" />
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

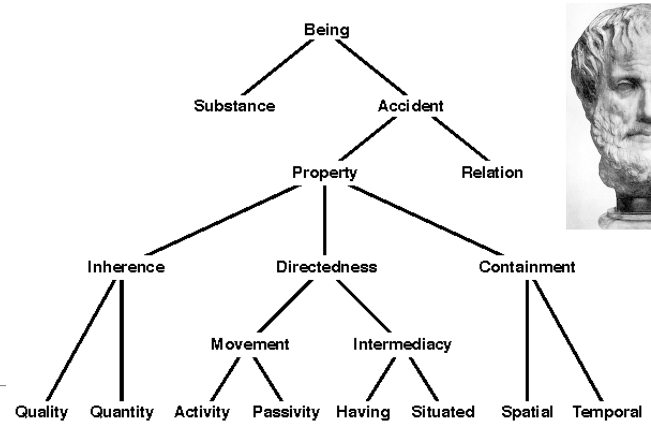
SCRIBE presentation by Rosario Usceda Sosa, Biplav Srivastava, Bob Schloss

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

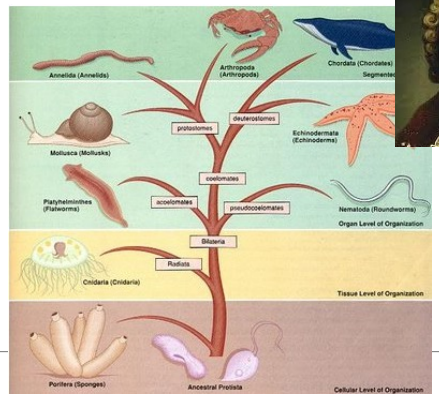
## What is an ontology, anyway?

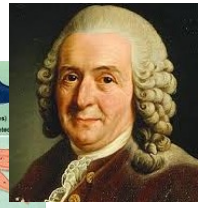
In Computer Science, “An ontology is a formal explicit description of concepts in a domain of discourse (**classes** (sometimes called concepts)), **properties** of each concept describing various features and **attributes** of the concept (slots (sometimes called roles or properties)), and **restrictions** 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





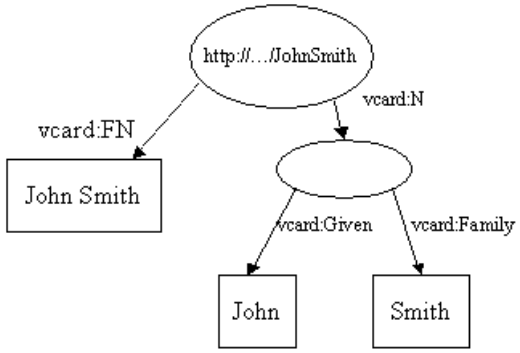



In a Smart City domain, we’re concerned with modeling the *city data* (city activity data, city departments, assets, KPIs), not the city itself (the full set of spatial and temporal relations between people and objects in the city) Ontologies help us to structure and reason about city *events*, *entities*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

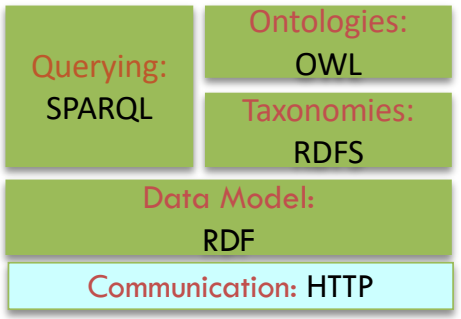
# OWL extends RDF…

RDF-schema
- Class, subclass
- Property, subproperty

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

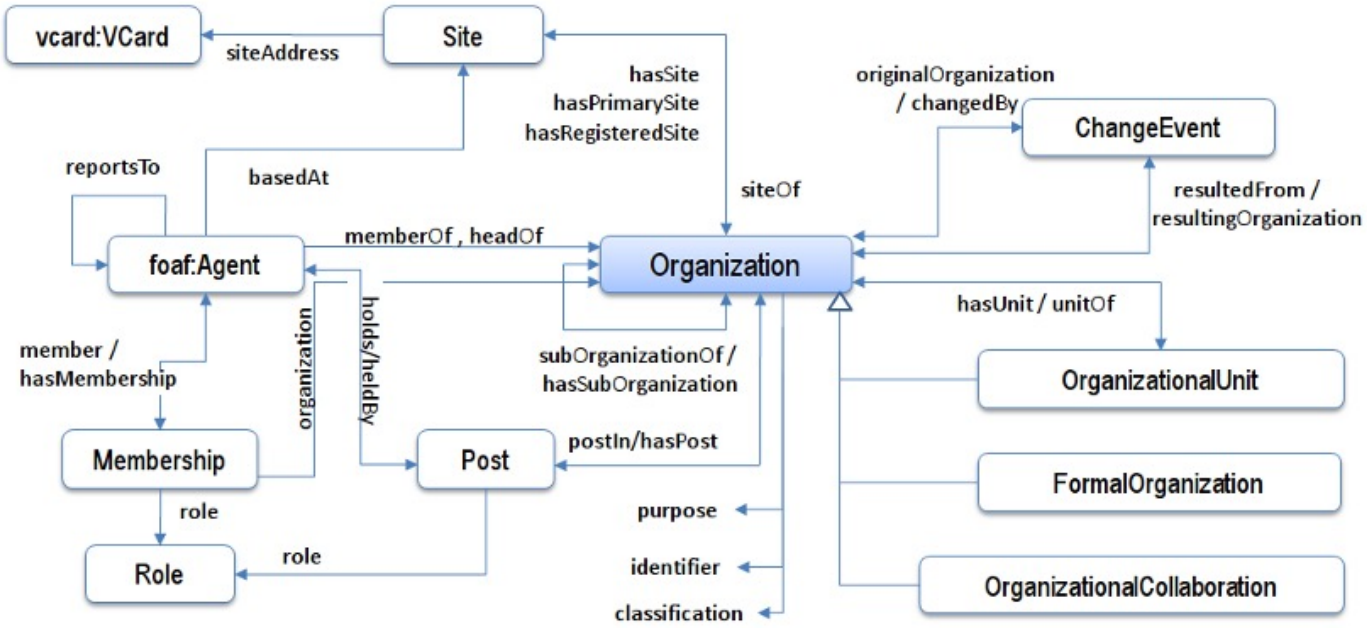
+ Combinators
- Union, Intersection
- Complement
- Symmetric, transitive

+ 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

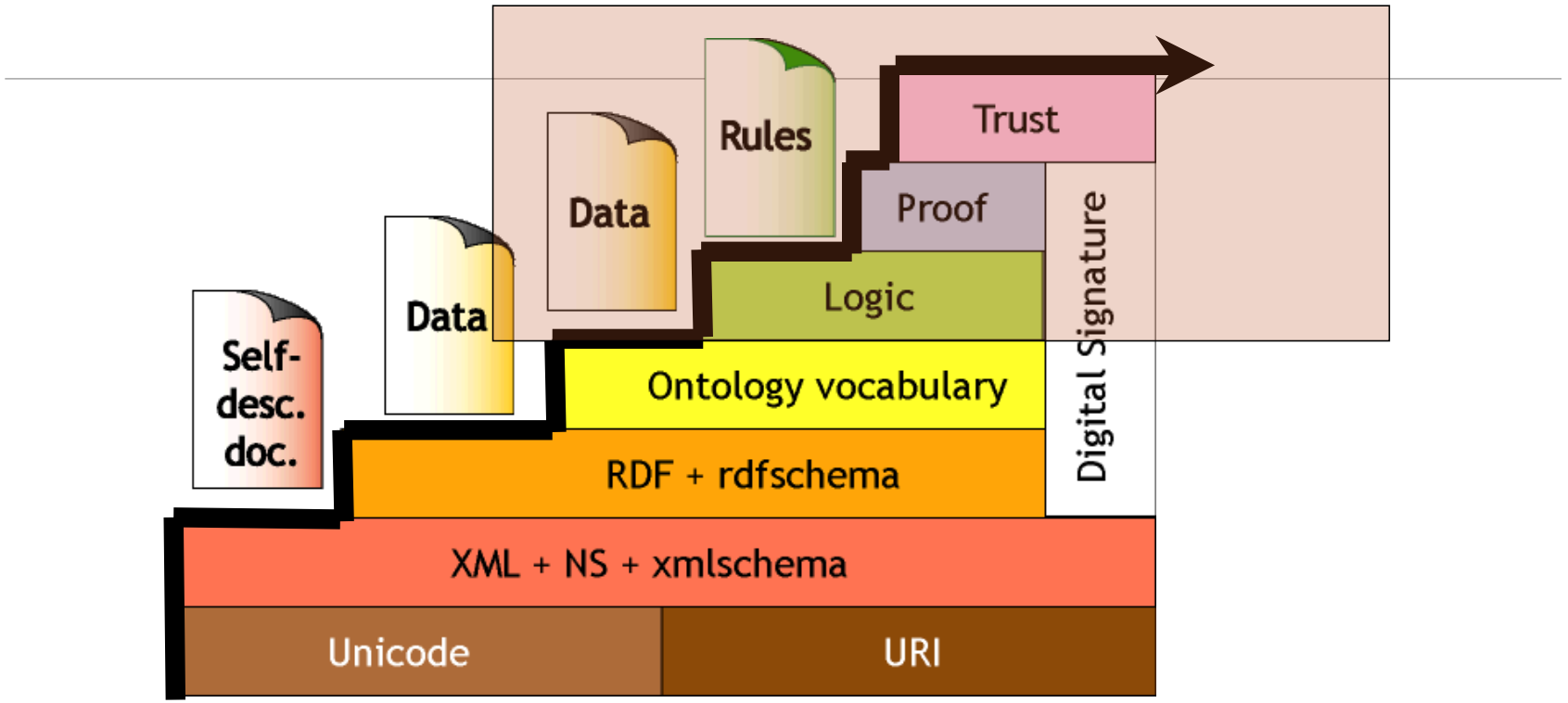
# Larger Ontology

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> ;
    .
```

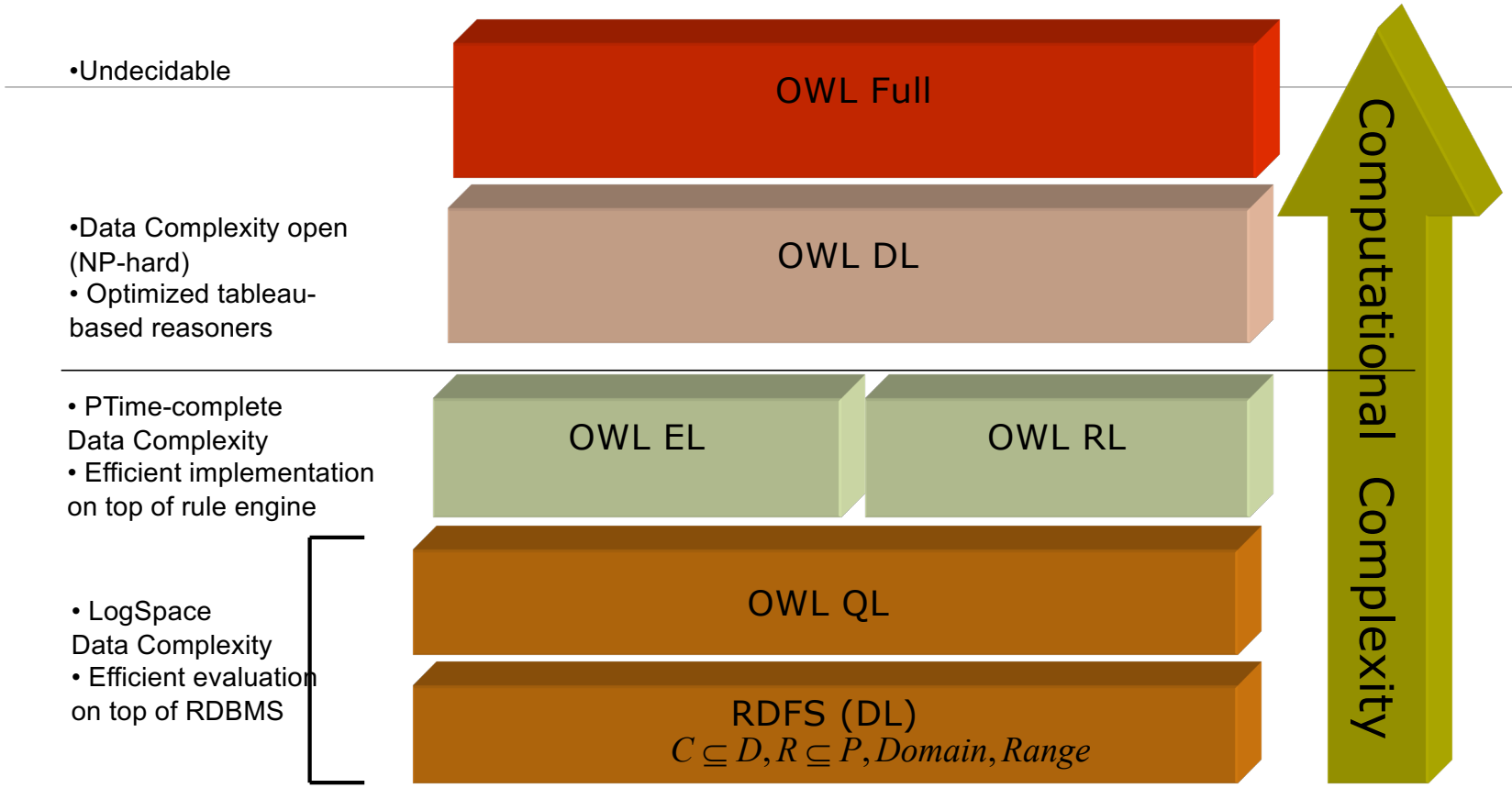

http://www.w3.org/ns/org

# Moving to the futureof 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 |
|---|---|---|---|---|
| **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) |

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.

**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

- ***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*

# 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
  - 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

# Concluding Section

# Course Project

# Discussion: Projects

- New: two projects, or
  - Project 1: model assignment
  - Project 2: single problem/ llm based solving / fine-tuning/ presenting result
- Old
  - Single problem of choice
  - Three sprints; solution to problem available via a chatbot that is built
  - Compare result with ChatGPT

# Project Discussion

1. Go to Google spreadsheet against your name
2. Enter model assignment name and link from (http://modelai.gettysburg.edu/ )

1. Create a private Github repository called "CSCE58x-Fall2024-<studentname>-Repo". Share with Instructor (biplav-s) and TA (vishalpallagani)
2. Create Google folder called "CSCE58x-Fall2024-<studentname>-SharedInfo". Share with Instructor (prof.biplav@gmail.com) and TA (vishal.pallagani@gmail.com)
3. Create a Google doc in your Google repo called "Project Plan" and have the following by next class (Sep 5, 2024)

Timeline
1. Title:
2. Key idea: (2-3 lines)
3. Data need:
4. Methods:
5. Evaluation:
6. Milestones
    1. // Create your own
7. Oct 3, 2024

# About Next Lecture – Lecture 3

# Lecture 4: Representing Knowledge

- World Knowledge: Physical, Beliefs, Probabilities
- Logic
- Inferencing