AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 580: Introduction to AI
CSCE 581: Trusted AI

Lecture 12: Machine Learning

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

3RD OCT 2023

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

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

Organization of Lecture 12

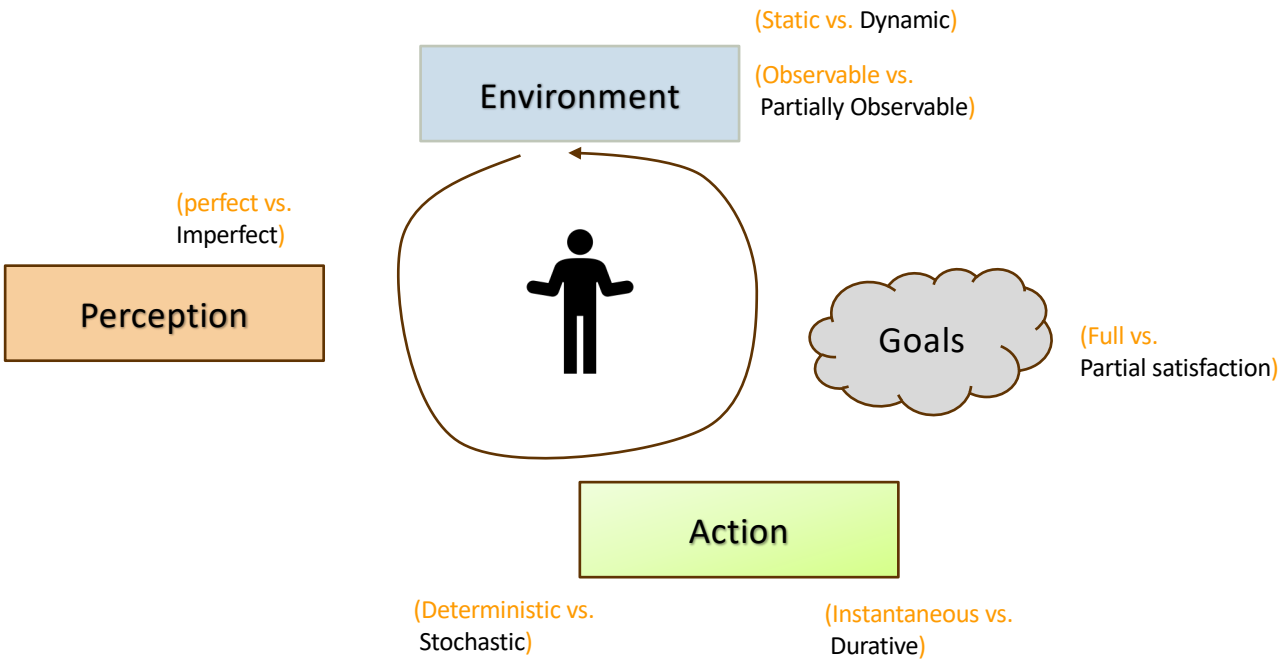
- Introduction Segment
  - Recap of Lecture 11
- Main Segment
  - Problem Settings
  - Data preparation and feature engineering
  - Solving classification problems
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 13
  - Ask me anything

Introduction Section

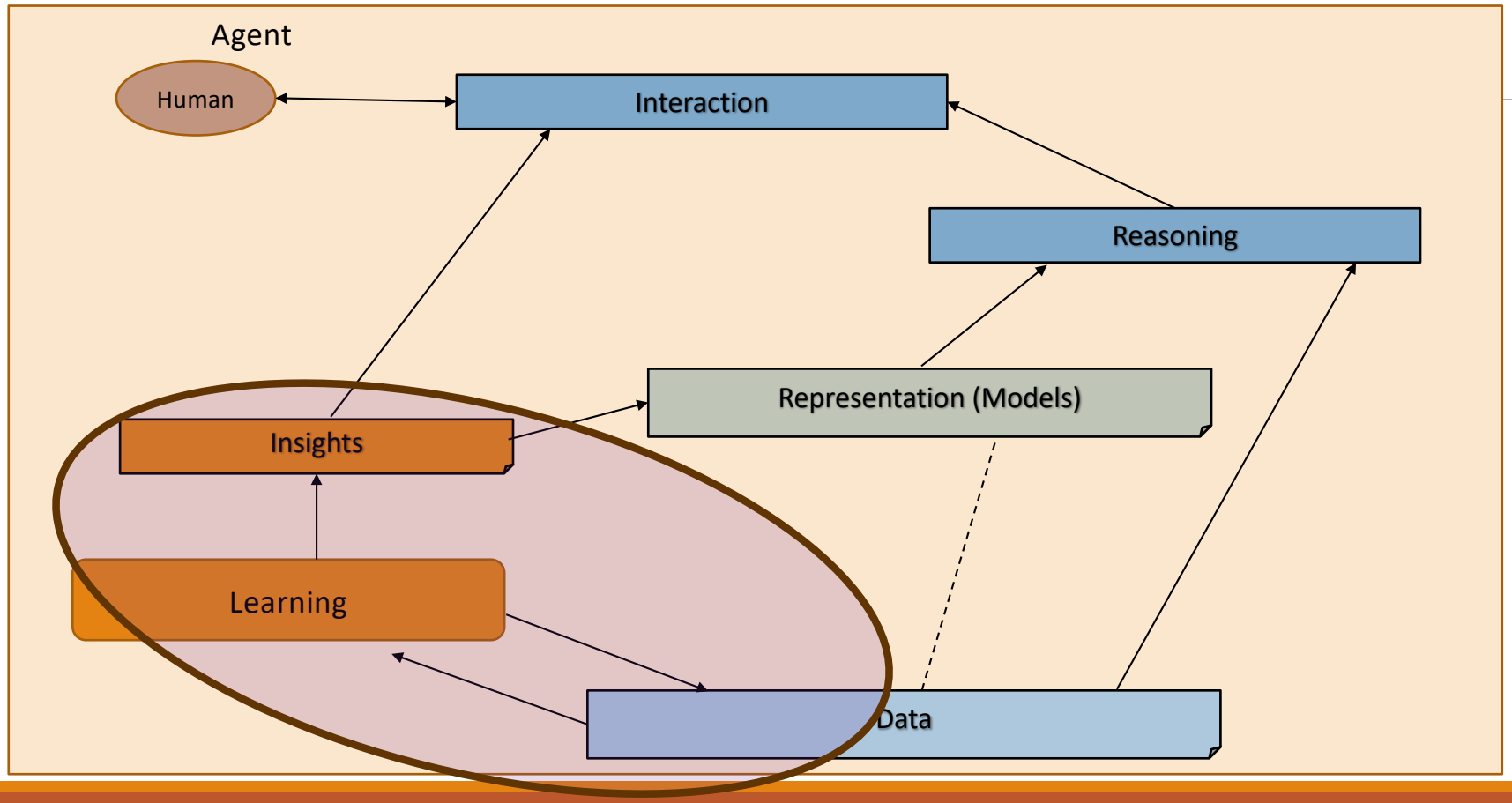
Recap of Lecture 11

- Constraint Satisfaction Problem
- Optimization Problems

Intelligent Agent Model



Relationship Between Main AI Topics



Where We Are in the Course

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

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

	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
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

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

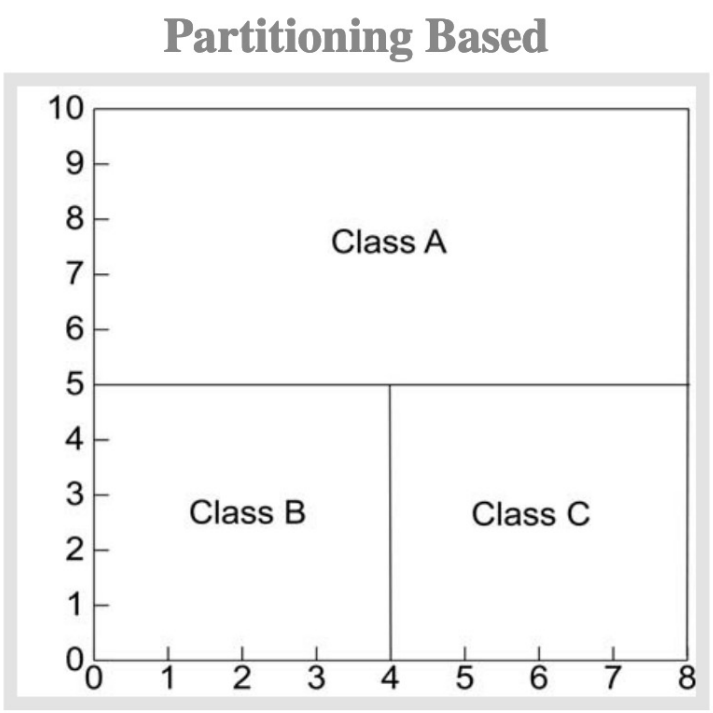
Reference: https://machinelearningmastery.com/difference-test-validation-datasets/
https://www2.seas.gwu.edu/~bell/csci243/lectures/classification.pdf

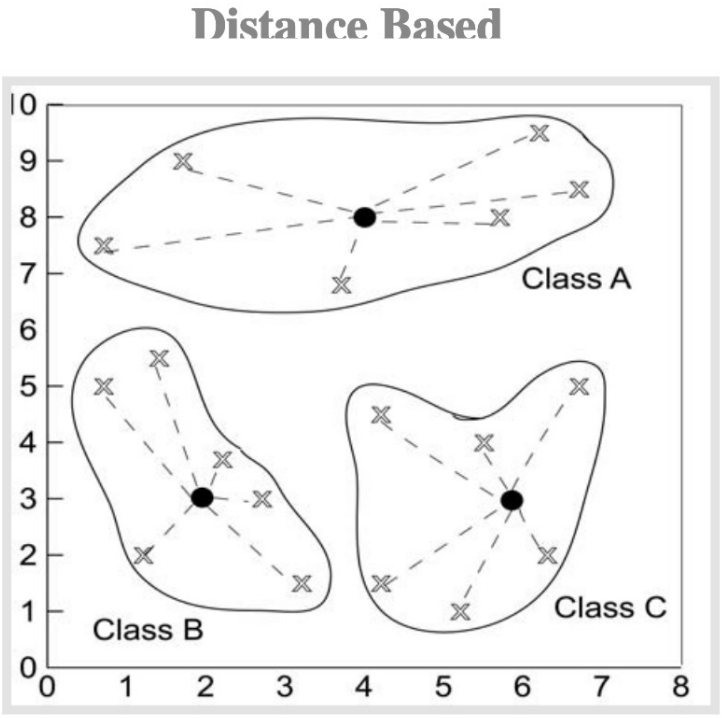
Sample Learning Task

- COVID-19 data

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

Methods for Classification

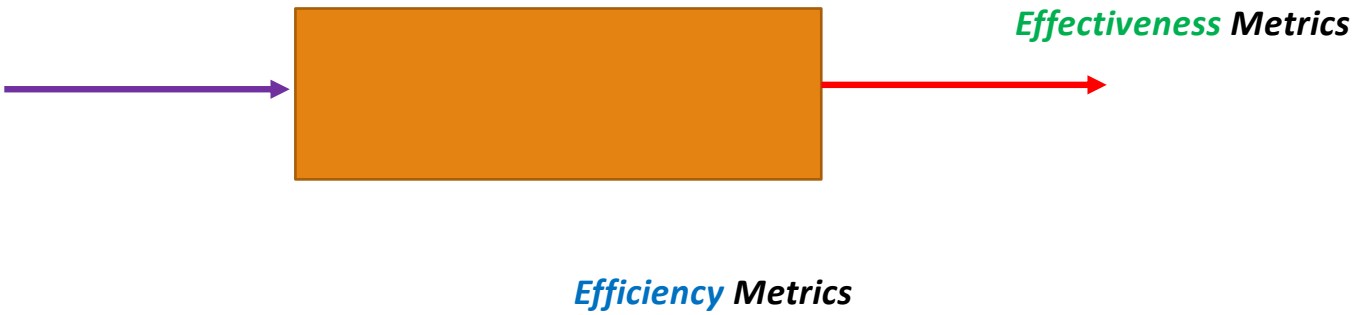




Source: https://www2.seas.gwu.edu/~bell/csci243/lectures/classification.pdf

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 ?*

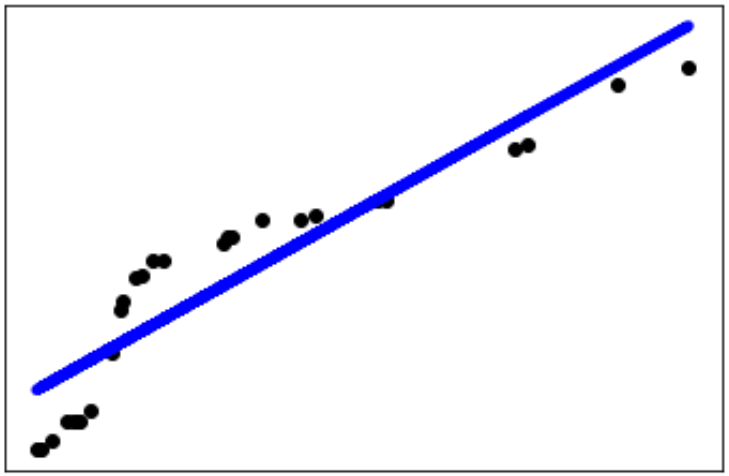
Comparing Classification Methods

- 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

Linear Regression



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

Metrics: Accuracy, Precision, Recall

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

Accuracy =
(TP+TN)/
(TP+FP+FN+TN)

Precision =
(TP)/
(TP+FP)

Recall =
(TP)/
(TP+FN)

F1 Score: *Harmonic Mean*

1/F1 = 1/Precision + 1/Recall

F1 = 2*(Recall * Precision) /
(Recall + Precision)

Reference and Demo

- Data: UCI Datasets - https://archive.ics.uci.edu/ml/datasets.php
- Tools:
  - Weka - https://www.cs.waikato.ac.nz/ml/weka/

Exercise: German Credit

Exercise and Code

- Linear Programming Methods
  - Link - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l16-optimal/Optimization.ipynb

Course Project

Project Discussion: What Problem Fascinates You ?

- Data
  - Water
  - Finance
  - …
- Analytics
  - Search, Optimization, Learning, Planning, …
- Application
  - Building chatbot
- Users
  - Diverse demographics
  - Diverse abilities
  - Multiple human languages

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

Project Discussion: Dates and Deliverables

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
- Nov 30, 2023
  - Project report due
- Dec 5 / 7, 2023
  - In-class presentation

Skeleton: A Basic Chatbot

- Run in an infinite loop until the user wants to quit
- Handle any user response
  - User can quit by typing “Quit” or “quit” or just “q”
  - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – “I do not know this information”.
- Handle known user query types // Depends on your project
  - “Tell me about N-queens”, “What is N ?”
  - “Solve for N=4?”
  - “Why is this a solution? ”
- Handle chitchat // Support at least 5, extensible from a file
  - “Hi” => **“Hello”**
  - …
- ***Store session details in a file***

Illustrative Project

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Discussion: Illustration

1. Create a private Github repository called "CSCE58x-Fall2023-<studentname>-Repo". Share with Instructor (biplav-s) and TA (kausik-l)
2. Create Google folder called "CSCE58x-Fall2023-<studentname>-SharedInfo". Share with Instructor (prof.biplav@gmail.com) and TA (lakkarajukausik90@gmail.com)
3. Create a Google doc in your Google repo called "Project Plan" and have the following by next class (Sep 5, 2023)

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Illustration: N-Queens

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
    - Method 1: Random solution
    - Method 2: Search – BFS
    - Method 3: Search - …
  - Human interaction: Develop a basic chatbot (no AI) as outlined
  - Deliverable
    - Code structure in Github
      - ./data
      - ./code
      - ./docs
      - ./test
    - Presentation: Make sprint presentation on Oct 12, 2023

Reference: Project Rubric

- **Project results** – 60%
  - Working system ? – 30%
  - Evaluation with results superior to baseline? – 20%
  - Considered related work? – 10%
- **Project effort**s – 40%
  - Project report – 20%
  - Project presentation (updates, final) – 20%
- **Bonus**
  - Challenge level of problem – 10%
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

Review: Regular Expression

Metacharacter	Explanation
^	Matches the starting position within the string
.	Matches any single character
[ ]	Matches a single character that is contained within the brackets
[^]	Matches a single character that is not contained within the brackets.
$	Matches the ending position of the string
*	Matches the preceding element zero or more times
+	Matches the preceding element one or more times
|	Separates choices

Regex	Matches any string that
hello	contains {hello}
gray|grey	contains {gray, grey}
gr(a|e)y	contains {gray, grey}
gr[ae]y	contains {gray, grey}
b[aeiou]bble	contains {babble, bebble, bibble, bobble, bubble}
[b-chm-pP]at|ot	contains {bat, cat, hat, mat, nat, oat, pat, Pat, ot}
colou?r	contains {color, colour}
rege(x(es)?|xps?)	contains {regex, regexes, regexp, regexps}
go*gle	contains {ggle, gogle, google, gooogle, goooogle, ...}
go+gle	contains {gogle, google, gooogle, goooogle, ...}
g(oog)+le	contains {google, googoogle, googoogoogle, googoogoogoogle, ...}
z{3}	contains {zzz}
z{3,6}	contains {zzz, zzzz, zzzzz, zzzzzz}
z{3,}	contains {zzz, zzzz, zzzzz, ...}

Example Source: https://cs.lmu.edu/~ray/notes/regex/

Implementation: Finding Words in Python

- Python has extended Regex specifications for convenience
- Useful for
  - Matching patterns
  - Information extraction
  - Content manipulation (e.g., substitution)
  - Error (e.g., spelling) correction

```
data = "The CSCE 771 course is taught at
        University this Fall!”
pattern = "[tT]+\w"
m = re.findall(pattern, data)
print(m)
```

['Th', 'ta', 'ty', 'th']

Details: https://docs.python.org/3/library/re.html

Regex Python Code Examples

- More regular expression examples
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l20-text-overview/WordLesson-Examples.ipynb

Lecture 12: Summary

- We talked about
  - Problem Settings
  - Data preparation and feature engineering
  - Solving classification problems

Concluding Section

About Next Lecture – Lecture 13

Lecture 13: Machine Learning

- Structured Data: Supervised Methods
  - Decision trees/ random forest
  - The variety of methods
  - Choosing a method that works

- Reading material:
  - "Which ML to Use" with title: Data-driven advice for applying machine learning to bioinformatics problems
    https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5890912/

  - "10 tips with title": Ten quick tips for machine learning in computational biology
    https://biodatamining.biomedcentral.com/articles/10.1186/s13040-017-0155-3