Fall 2023

AME 505:
Engineering Information Modeling:
Machine Learning for Engineering Informatics

Department of Aerospace & Mechanical Engineering

University of Southern California

COURSE SYLLABUS

Instructor: Prof. Yan Jin

Professor of Aerospace and Mechanical Engineering

Office: Room 412C, Olin Hall of Engineering

Email: yjin@usc.edu

http://impact1.usc.edu/yjin

Course Section:	28864R (campus) 28865D (den)
Course Unit:	3 Units
Prerequisite:	Graduate Standing
Class Hours:	Tuesdays 6:40 pm – 9:20 pm
Class Location:	RTH-109 & DEN@viterbi
Office Hours:	Tuesdays 11:30 -12:30 or by appointment @ OHE-412C & Zoom
Teaching Assistant:	TBD

AME505:
Engineering Information Modeling:
Machine Learning for Engineering Informatics

Fall 2023

Course Description and Objectives

Engineering informatics is an interdisciplinary field that applies computing and information science to advance the knowledge and practice of engineering design, analysis, and manufacturing. As the increasingly complex practice of engineering transcends human experiences and becomes increasingly dependent on information and data, there is a strong need for new methods and capabilities to guide engineering activities and manage and transform data into helpful engineering insights. Engineering informatics involves the development of novel methods and algorithms for enhancing engineers' creativity, automating engineering processes, and developing novel future systems.

The recent progress in AI/ML (*artificial intelligence* and *machine learning*) theories and technologies has opened ways to solve many complex engineering problems that involve human intelligence. To compete in today's relentless market and contribute to engineering advancement, an engineer must have a good grasp of contemporary AI/ML technologies, identify and model engineering problems in AI/ML-relevant terms, and develop quality AI/ML solutions to these engineering problems. Information modeling and AI/ML technologies are fundamental for developing such solutions. In this course, AI/ML theories and techniques are introduced and applied, including object-oriented modeling, machine learning, and data analysis, to solve engineering problems that usually involve human intelligence. Developing computational thinking and fostering creativity are also two intended goals of this class. Specific course objectives are for students to:

- *Understand AI/ML principles and algorithms*: We will introduce general backgrounds of AI/ML and then provide an overview of how engineering information can be semi-formally modeled with symbols, lists/sentences, and objects. After that, the details of AI/ML principles and algorithms will be presented and discussed, together with some engineering examples, through a three-phase structure: fundamentals (supervised, unsupervised learning, and reinforcement learning), basic skills (Python-based system development), and advanced topics (genetic algorithm, support vector machines, Bayesian learning, and generative pretrained transformers).

- *Develop basic skills by building a machine learning based engineering support system*: Students will learn basic AI/ML and programming skills using *Python* and other modeling and machine learning tools for classroom exercises and term projects. The hands-on experience will be helpful for students to digest the principles and algorithms and develop system solutions. Note that this class will teach Python and does NOT require students to have a Python background.

- *Understand the state of the art of AI/ML technologies*: Throughout this course, we will discuss state-of-the-art artificial intelligence, machine learning, and data analysis methods and their application case studies. We will examine specific engineering domains and fitting AI/ML algorithms that can be applied to improve engineering practice. Guidance will be provided so students can explore beyond the scope of this class based on their interests.

- *Apply the learned knowledge and skills to solve engineering problems*: Students will form teams for their term project from the beginning to the end of this course. Each team will prepare a project proposal, develop a proof-of-concept AI/ML computer system, and submit a final technical report of their project. The project teams will present their term project along with a live system demo to the class at the end of this course.

Course Structure

This course covers four major components: modeling methods, fundamentals of AI/ML, programming & system building techniques, and advanced topics. The course structure is designed to include the following modules to make the learning process more effective and efficient.

Part 1 (1 week): Basic modeling methods

This module aims to introduce basic concepts and methods of information modeling.

- What is engineering information modeling?
- What is symbolic modeling?
- What is object-oriented modeling, and why is it so important?

Part 2 (6 weeks): Fundamentals of AI/ML – Theories and Methods

This module introduces engineering-relevant machine learning and data analysis theories and practices.

- What are machine learning and data analysis; what are their different types?
- What are some of the techniques for supervised learning and unsupervised learning?

- What is deep learning, and when to use it?
- What is big data? What to do with Big-data and Small-data?
- What is reinforcement learning, and how to apply it?

Part 3 (3 weeks): Python Programming Techniques

This module offers basic Python programming techniques for students to apply their learned knowledge and techniques to develop AI/ML system solutions.

- What are Python and PyCharm programming environments?
- What are the basic programming techniques using Python?
- How can one build an application system in a Python programming environment?
- How can one build graphical user interfaces (GUIs) quickly?
- What are programming packages (e.g., machine learning) available?

Part 4 (4 weeks): More on Machine Learning– Algorithms & Applications

This module will introduce a genetic algorithm, support vector machines, Bayesian learning, and generative pretrained transformers (GPT), and their applications will be discussed.

- What is evolutionary computing?
- What is SVM (support vector machine), and where does it perform better?
- When do we need Bayesian learning, and why?
- How do GPT and ChatGPT work?
- What are some of the applications of these AI technologies in engineering?

Prerequisite

Graduate standing. No restrictions on graduate students. Senior students may also take this class after receiving approval from the instructor.

Course Work

Classroom Lectures:

Weekly lectures will be offered on Tuesdays. Students must complete reading assignments indicated on the schedule page before each class. Usually, the weekly 3-hour classroom lecture is divided into two parts. During the first 80 minutes, the instructor will present and discuss the contents outlined on the schedule page. In the second 70 minutes, students will work *in groups* to discuss the topics introduced by the instructor and complete the "classroom exercises" by applying learned methods to small engineering problems (in the first several weeks) and their team project problems (as the course progresses). Active participation in classroom exercises is strongly required for all students.

Quizzes:

Online quizzes will appear after each lecture on the Desire2Learn class website and are due before the next class session. Quizzes usually cover the materials presented and discussed in the last class.

Homework:

Each homework assignment usually has 2-3 short questions. The questions are usually about (1) the content of the previous lecture and (2) the reading assignment for the following lecture. The homework is intended to help students digest critical concepts learned from the lectures, assimilate the reading materials, and organize their thoughts for class discussion and future practice.

Mid-term Exam:

The mid-term exam will be open-book. Exam problems will be similar to, and more comprehensive than, the homework questions.

Term Project:

Students will form teams of 4-5 people for the term project. Each team will work together to propose a project topic by choosing a specific engineering application problem. Teams will develop particular information models and machine learning and data analysis algorithms, build system components including GUI (graphical user interface), and compose a demonstratable system to solve their selected project problem. The term project will allow students to review and apply the theories, methods, and techniques of modeling, machine learning, and data analysis they learned from the class.

Course Materials

- Required: AME505 Course Reader: This will be available on the course website.
- Others (will be added).

Grading Requirements

Quizzes: 15%

A total of nine (9) quizzes will constitute 15% of the overall grade. Online quizzes will appear after each lecture on the Desire2Learn class website and are due before the next class session at 6:30 pm.

Homework: 30%

Ten (10) homework assignments will constitute 30% of the overall grade, each amounting to 3%. Each homework assignment has 2-3 short questions and/or one small modeling or analysis problem. Questions can be open-ended. Thoughtfulness, clarity, conciseness, and incisiveness are required.

Midterm exam: 25%

The midterm exam is open-book and open-note, limited to the materials discussed in the classroom lectures, quizzes, homework assignments, and team projects. Questions will be similar to, but more comprehensive than, the homework questions and system development problems. There will be query questions and small modeling and analysis problems involved.

Term project: 30%

Note that 30% of your semester grade is based on the result of your term project, which is a team effort. All project work done by a team is first given a team grade. This team grade is then weighted for each student, based on confidential peer evaluations by all team members at the end of the semester according to the following scheme.

Each student will be asked to fill out a questionnaire, which rates every team member (including him/herself) for the percentage participation in the team project, from 1 (10% or less participation) to 10 (100% participation). The evaluations are averaged to find each student's involvement. The project grade of each team student will be weighted based on his or her percentage of participation.

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriatesanctions/. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct/.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu/ or to the Department of Public Safety http://capsnet.usc.edu/department/department-publicsafety/online-forms/contact-us. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report or can initiate the report on behalf of another person. *The Relationship and Sexual Violence Prevention & Services* https://studenthealth.usc.edu/rsvp/ provides 24/7 confidential support, and the sexual assault resource center webpage https://sarc.usc.edu/reporting-options/ describes reporting options and other resources.

Students with Disabilities

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to the instructor as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. Website and contact information for DSP: http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html,(213) 740-0776 (Phone) (213) 740-6948 (TDD only), (213) 740-8216 (FAX), ability@usc.edu.

Emergency Preparedness/Course Continuity in a Crisis

In case of a declared emergency, if travel to campus is not feasible, *USC Emergency Information* http://emergency.usc.edu/ will provide safety and other updates, including ways in which instruction will be continued by means of the blackboard; teleconferencing, and other technology.

Course Schedule

Date	Lecture	Class Exercise	Readings	Quiz Due	Work Due
8/22	Course Intro & Intro to AI and engineering applications	CE#0: Teaming CE#1: Discuss engineering apps	Course Syllabus		
8/29	Object-oriented modeling & Rule-based systems	CE#2: Vending machine modeling CE#3: Object-oriented modeling CE#4: Programming in CLIPS	Chap1: Object-O Modeling Chap2: RBS	Quiz#1	HW#1
9/5	Supervised learning: Nearest neighbor, decision tree, etc.	CE#5: Data analysis tool intro CE#6: Data analysis with kNN & DT	Chap3: Into 2 Data Mining Chap4: Classification	Quiz#2	HW#2
9/12	Neural network and Deep learning	CE#7: Data analysis with neural net	Chap5: Introduction to Neural Network	Quiz#3	HW#3
9/19	Unsupervised learning: K-mean hierarchical clustering, etc.	CE#8: Clustering data and finding insights	Chap6: Cluster Analysis (k-Mean & Hier-clustering)	Quiz#4	HW#4
9/26	Unsupervised learning: collaborative filtering, etc.	CE#9: Discuss the team project topic	Chap7: Collaborative Filtrg Chap8: Assoc. Analysis	Quiz#5	HW#5
10/3	Reinforcement learning: Q-learning, deep RL	CE#10: Applying reinforcement learning	Chap9: Reinforcement Learning	Quiz#6	HW#6
10/10	Midterm Exam				Project proposal
10/17	Python A: Concepts, structure	CE#11: Install and use PyCharm	Python, PyCharm doc	Quiz#7	HW#7
10/24	Python B: Common and machine learning packages	CE#12: Programming in Python	Python, PyCharm doc		
10/31	Python C: GUI/QtDesigner and build apps	CE#13: Program your project	Python, PyCharm doc		
11/7	Evolutionary computing & Genetic algorithm	CE#14: Project Briefing	Chap10: Genetic Algorithm		Progress Report HW#8
11/14	Support vector machine, etc.	CE#15: Project Q&A (1)	Chap11: SVM	Quiz#8	HW#9
11/21	Bayesian Learning	CE#16: Project Q&A (2)	Chap12: BL	Quiz#9	HW#10
11/28	GPT and LLMs (large language models)	Project Presentation	Chap13: GPT & LLM		
12/7	Final Project Report PDF file due 11:59 pm				Final-Project Report

CE: Classroom Exercise