EN.605.649 section 81 Syllabus

Introduction to Machine Learning

Course Information

Course Information:

Introduction to Machine Learning
EN.605.649.81 (3.0 Credits)
Summer 2024 [AE Summer 2024]

Description
Analyzing large data sets (“Big Data”), is an increasingly important skill set. One of the disciplines being relied upon for such analysis is machine learning. In this course, we will approach machine learning from a practitioner’s perspective. We will examine the issues that impact our ability to learn good models (e.g., inductive bias, the curse of dimensionality, the bias-variance dilemma, and no free lunch). We will then examine a variety of approaches to learning models, covering the spectrum from unsupervised to supervised learning, as well as parametric versus non-parametric methods. Students will explore and implement several learning algorithms, including logistic regression, nearest neighbor, decision trees, and feed-forward neural networks, and will incorporate strategies for addressing the issues impacting performance (e.g., regularization, clustering, and dimensionality reduction). In addition, students will engage in online discussions, focusing on the key questions in developing learning systems. At the end of this course, students will be able to implement and apply a variety of machine learning methods to real-world problems, as well as be able to assess the performance of these algorithms on different types of data sets. Prerequisite(s): EN.605.202 – Data Structures or equivalent.

Department: PE Computer Science
College: Engineering and Applied Science Programs for Professionals

Instructor Information:

Instructor

John Sheppard
jsheppa2@jhu.edu

Teaching Assistant:

All assignments will be graded by Will Jardee (wjardee1@jh.edu), who is a PhD student at Montana State University and a member of Dr. Sheppard's lab. Feel free to reach out to them via email.

Note that any grading appeals should first be directed to the grader within one week of receiving the graded assignment. After receiving a decision from the grader, should an additional appeal be required, feel free to reach out to the instructor, again within one week of receiving that decision.

Course Location:

Online

Communication Policy:

Dr. Sheppard prefers that students contact him via email at jsheppa2@jh.edu or through the Canvas email system. Dr. Sheppard will make every effort to respond to your email within 48 hours or earlier. If necessary, you may call him at 406.994.4835. Please be sure to leave a message if there is no answer.

If an issue is urgent, the student should include "URGENT:" at the beginning of the subject line of the email and Dr. Sheppard will respond as soon as is practical. In urgent situations, the student is also encouraged to call Dr. Sheppard during normal business hours.

Office Hours:

Open (public) office Hours will be held at least once per week at a time set during the first week of the semester. Office hours sessions will use Zoom. The office hours "link" will be available on the left-hand menu within the course site. Open office hours sessions will be recorded. In addition, 20-minute one-on-one office hours sessions via Zoom can be scheduled. Available times will be posted in the calendar the week before they are to occur, where students will be able to sign up for one of the open spots. One-on-one office hours will not be recorded.

Course Structure:

Details on the course structure can be found in the Course Outline. Each course module runs for a period of seven (7)days, i.e., one week. Due dates for readings and other assignments are referred to by the day of the module week inwhich they are due. For example, if a reading assignment is to be completed by Day 3 and the module started onMonday, then the reading assignment should be completed by Wednesday or the 3rd day of the module. Please refer tothe Course Outline for the specific start and end dates for each module in this course.

IMPORTANT: The course design has recently been updated to provide more opportunities for students to manage their time. In particular, all programming assignments will be allotted three weeks for completion; however, you will be required to submit your code by the end of the second week. The third week is designed to be used for creating your video and writing your paper. That said, if you need to update your code, you may do so; however, whether you update or not, resubmit your code at the end of the third week.

Course Topics:

1. Non-parametric Learning
2. Clustering
3. Bayesian Learning and Decision Trees
4. Ensembles
5. Dimensionality Reduction and Rule Learning
6. Representation Learning and Linear Models
7. Feedforward Neural Networks

8. Other Neural Networks
9. Deep Learning
10. Reinforcement Learning
11. Temporal Difference Learning
12. Deep Reinforcement Learning

Course Goals:

To develop broad understanding of the issues in developing and implementing machine learning algorithms and systems, especially as they related to modern, data-intensive problems.

Course Learning Outcomes (CLOs):

- Determine the inductive bias of learning methods and how that bias potentially affects learning performance.
- Assess the impact of the bias-variance dilemma and no free lunch in utilizing machine learning methods.
- Differentiate between classification and function approximation (regression) in learning.
- Manage learning complexity through unsupervised learning methods.
- Manage model fit through regularization and validation data sets.
- Assess the role of linearity in model selection.
- Implement and apply a variety of machine learning methods to real-world problems.
- Assess the empirical performance of machine learning algorithms on different types of data sets.

Required Text and Other Materials

Textbooks:

Required

Alpaydin, E. (2020). *Introduction to machine learning*(4th ed.). Cambridge, MA: The MIT Press. ISBN 9780262043793. (Note: Earlier editions should be used at your own risk.)

Prior editions of this book are strongly discouraged and may be used only at your own risk.

Optional

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill. ISBN-10: 0070428077; ISBN-13: 9780070428072.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY: Springer, ISBN-10: 0387310738; ISBN-13: 9780387310732.

Access to textbooks via the JHU Libraries:

EP students may access electronic versions of textbooks through the Sheridan Libraries. Instructions on how to search for available textbooks are accessible through this link: Browse Electronic Textbook Instructions

Other Materials & Online Resources:

Speakers/audio output (or headsets) and microphones are required for this course. A webcam is required for use during office hours.

Technical Requirements:

You should refer to General Technical Requirements for guidance on system requirements. Access support resources from the **Help** menu if you encounter any technical issues.

Evaluation and Grading

Student Coursework Requirements:

Course Expectations

This is a very high workload course and is designed as a graduate-level computer science course. It has also been designed to prepare students interested in a more technical/research-based experience in the design of their degree programs. All students, but especially those in programs other than the computer science MS or postgraduate certificate program, should therefore, be aware of the following expectations.

- All students are expected to be able to program at a professional and graduate level. As such, no programming-specific advice or debugging will be done by either the instructor or the course grader. It's up to the individual student to have developed the necessary skills to be able to complete large, complex programming assignments commensurate with these expectations.
- Students have reported a wide range of time commitments required for this course, ranging from 10 hours per week to 30 hours per week. The intent is for each programming assignment to require no more than 10 hours per week, but higher time commitments are not uncommon. Much of the time required is directly related to the programming experience of the student.

- All software written for this course is expected to be designed and documented as if done by a professional/graduate-level programmer, employing sound design principles and citing any/all resources used in the creation of the software.
- Software developed for this course may not be posted in a public code repository or on a public website ever, including after the course is completed. To do otherwise will be considered suborning academic conduct, which in itself is a form of academic misconduct.
- The various degrees offered by EP and AAP require different foundational work prior to taking other graduate courses in their respective programs. The key foundational requirement for this course is 605.621 Foundations of Algorithms. Note that the Data Science and Artificial Intelligence versions of these courses, while accepted as pre-requisites, may not be sufficient to be fully successful in this course.
- The course has been designed to provide a formal and practical introduction to the subject of machine learning. As such, a fair amount of mathematical maturity is expected of students taking this class. The official pre-requisites are the CS foundation courses and CS mathematics pre-requisites for the MS degree. That said, students would be well-served if they have completed the following math courses: multi-variate calculus, probability and statistics, discrete mathematics, and linear algebra. All of the lectures assume familiarity with these subjects.
- Should any issues arise (personal, professional, or technical), it is expected that students will act professionally and reach out to the instructor as soon as possible regarding these issues. Without prompt communication, it is impossible for the instructor to be able to work with the student to help to resolve the issues positively.
- Under no circumstances will time management issues be regarded as an acceptable excuse when impacting the completion of work in this course.
- The discussion exercises (small group and muddy point) are intended to provide a means for students to explore topics in depth beyond the lecture material in the course. It is expected that all students will participate in these exercises actively and substantively. The level of return from these exercises will be related directly to the level of effort exerted in completing the exercises.
- The quizzes are designed to fall somewhere between summative and formative assessment. They are graded exercises, but a) all students get two attempts and b) the overall impact of the quizzes on the final grade is relatively low. Even so, it is expected that all students will take them seriously since they also provide a sense to both the student and the instructor of where there are issues and points of confusion. That said, the questions posed are not trivial, so students should take time to prepare for them and to make sure they understand the questions and possible answers.
- The course grader is just that, a grader. They are not a teaching assistant in the traditional sense of the word. Even so, there may be circumstances where the grader fills in for the instructor in things like leading office hours. Therefore, all students are expected to treat the grader with respect as if acting as a representative or proxy for the instructor.
- There is a significant writing requirement for this course. The intent is for students to be able to craft a report on the experiments they conduct with each of the programming assignments. The expectation is for students to gain experience to prepare them for future courses that may incorporate research-based experiences.
- Students are expected to do all of their own work without the help of generative AI tools such as OpenAI's ChatGPT or Microsoft's Copilot. Therefore, it is also expected that, even if this policy is violated, all students will act with integrity and disclose any and all use of generative AI tools, or other programming resources, in the completion of their assignments. To do otherwise will be regarded as plagiarism, which is a form of academic misconduct.

Programming Assignments Requirements

The programming assignments are designed to give you experience implementing key machine learning algorithms from scratch. You will implement the algorithms and then test their performance on several data sets from the UCI Machine Learning Repository. For these assignments, you are required to submit source code, short videos that demonstrate proper functioning of the code, and a brief paper describing of the results of your experiments.

You may use any of the following higher-order programming language you wish, Java, Python, or C#; however, you are ***not permitted*** to use available machine learning libraries such as Matlab toolbox, WEKA, RapidMiner, scikit-learn, TensorFlow, etc. You are not permitted to seek out and use any code found in online code repositories that appear to be similar or the same as the programs assigned in this course. You are not permitted to use the following languages for

implementing the algorithms, but you may use them to support analysis of the results: SQL, Matlab, Maple, Mathematica, R. All algorithms must be implemented from scratch by you. Basic libraries for managing data structures and fundamental math operations (e.g., NumPy or Pandas) are permitted.

To facilitate the grading of programming assignments, please adhere to the following:

- All code is required to be submitted one week before the assignment is due but is not required to be fully commented at that time. Code may be revised during that the final week of the assignment. Fully commented code must be resubmitted when the full assignment is due. Failure to provide the first code submission will result in getting a zero on the "code" portion of the assignment (i.e., 10%).
- All source code shall be submitted as part of the assignment when your programming assignment is submitted. A separate file containing all of the elements for the programming assignment will be uploaded. You are permitted to use notebooks (e.g., Jupyter), but you are not permitted to upload notebooks for your submission. Only source code should be uploaded for the programming part of the assignment; external libraries and deprecated should not be included.
- The source code shall be zipped into an archive file where the name of that archive file will be <assignment#>.zip or<assignment#>.tar.gz (or whatever is appropriate to your compression software). The "#" should be replaced with the number of the assignment. Note that there will be four assignments.
- All source code must be fully commented. It is common practice to use JavaDoc-type comments at the start of a method that explains what the method is doing. Inline comments are also needed to explain the logic of the code. In addition, the code should use method and function names that are "self-documenting." For example, use "index" rather than "i" or "data[ ]" rather than "a[ ]".
- Results of running your programs are required of all programming assignments. Results will be provided by creating a video of your program(s) running, where the video for most assignments is limited to five minutes. Do not walk through the code in the video. Focus, instead, on functionality, as outlined in the assignment. Note that we will view nothing beyond the indicated limit.
- Your video may be submitted as a separate .mp4 file, or a link to your video on a streaming service (e.g., YouTube) may be provided. Note that broken links or security settings that prevent access will be treated as if the video was not submitted, so be sure to make sure the grader will be able to view the video.

The report you provide should be done with a word processor or Latex. We recommend using Overleaf, but this is not required. Use the equation and pseudo-code editing capabilities of whatever tool you use. As this is a graduate level course, it is expected that you follow appropriate conventions related to incorporating mathematics, pseudocode, and figures into the paper. Make sure you submit a PDF of your report. MS Word, Open Office/Libre Office, and Latex files will not be accepted. **Submit your report separately, i.e., do NOT include in in your zip file.**

Your report must include the following: [1]

- Title, Author
- Abstract
- Problem statement and testable hypothesis
- Brief description of algorithms studied
- Experimental design
- Description of any tuning process applied
- Results
- Analysis (including statistical analysis)
- Conclusions
- References (only need to include those elements used that go beyond the course content. If needed, references should include any external resource used)

[1] A sample report is provided in Canvas. Note that the sample report specifies requirements for a more formal reportthan what is specified here. You are only required to satisfy the requirements above.

Participation Grading Criteria

Active student participation is an essential part of any online course. Therefore, part of the student's grade (30%) will bebased on class participation. There are two components to the class participation grade – muddy posts/responses and small group discussions.

As a reminder, the use of generative AI (e.g., Gemini or ChatGPT) is ***not permitted*** when participating in any of the discussion exercises. The point is for you to express what you know, not what an online AI tool knows.

CAUTION: Be advised that you will be assigned to two *different* groups – one for muddy point exercises and one for small group assignments. Members of these groups are *not the same*, and either or both may change as the semester progresses.

Muddy Point/Response

At various points in the class, students will be required to post a "Muddiest Point" message to the muddy point discussion forum associated. Specifically, the student shall post a comment that identifies a part of the module that was particularly confusing, thus needing clarification. This must be done by Day 3 of the week when the discussion takes place but can cover anything since the previous muddy point exercise. Note, however, that only one point/question is to be posted. Students will be paired with one or two other students (called muddy buddies), and one of the partners will post a clarifying response as a reply to the muddy point within two days of the initial posting (i.e., by Day 5). This response must constitute a serious, substantive attempt to answer the question posed in the muddy point and will be graded accordingly. Simply referring to an external website (e.g., Wikipedia) is not sufficient. The responder must demonstrate that they have attempted to gain a solid understanding of the answer. Thus, students will be evaluated based on timeliness in posting the Muddiest Point as well as their ability to provide clarifying responses to their partner's Muddiest Point. Later, the instructor will add additional clarifying information if necessary.

Types of questions that are not accepted for muddy points include the following:

- Asking to compare/contrast algorithms to determine when to apply which. That should come from the course content and gaining an understanding of the underlying inductive biases.
- Asking for a walk-through or detailed example of how an algorithm works. Questions focused on how algorithms work are fine, but they should be restricted to understanding the logic of how they work. The person responding is free to provide examples but should not be expected to.
- Asking about material that goes beyond the scope of the class. Such topics are for the advanced class to address.
- Asking speculative or research-oriented questions. Since such questions often lack foundation or require opinion, they are not suitable for assisting in understanding the basic content of the course.
- Asking a question that comes from the exercises of any textbook (especially the textbook assigned for the class).

For grading, each muddy response will be scored based on timeliness, completeness, and correctness (1: on time, complete, and correct, 0.5: late, incorrect, or answers a question not asked, 0: unacceptable). Incomplete, incorrect, or deficient answers will be graded using a 0.6–0.9 multiplier; however, once the instructor responds with an answer, no credit will be provided for any subsequent responses. Each muddy post will also receive one of three scores (1: on time and substantive, 0.5: late or not substantive, 0: no post or unacceptable). The response is weighted 50% more than the initial post. Each week's muddy score is calculated as the post score plus the 1.5 times the response score, and this total divided by 2.5 to obtain a percentage. This score (out of 100) is what will be posted in Canvas. All of the muddy scores are averaged for the final muddy point grade. As mentioned above, a muddy point may contain one, and only one, target question to be addressed. If multiple questions are posted, a penalty will be applied to the muddy point part of the grade. Furthermore, the muddy buddy responding may then choose which question to answer and is not required to address all of them.

When there are groups of three people, a “round robin” response policy is enforced. This means that everyone needs to be the primary responder to one other person in the group. The specific order of response (i.e., who responds to whom) will be indicated during each muddy point exercise.

As a reminder, generative AI tools such as ChatGPT or Gemini are not permitted to be used when answering a muddy post. The response must be written entirely by the person responding to the muddy point.

The muddy point/response part of the course is worth 15% of the final grade.

Small Group Discussions

In addition to the muddy point exercises, at various points in the class, an open discussion question will be posted in the main discussion forums. Each student will be grouped with two or three other students and assigned to a "Group" within Canvas. During the week, the group is to engage in an ongoing discussion on the question posted. Each student is required to post substantively at least five times, and these posts must occur on at least three different days. Non-substantive posts (e.g., "I agree," or "I need to think about that more") will not be counted. Substantive posts must address the original prompt or response to the initial prompt. Note that a single response should not address the entire prompt.

During each discussion, each substantive post will receive 1 point (up to 5 points total), and each day posted will receive1 point (up to 3 points total). Thus, full credit will be 8 points. The score posted in Canvas will be a percentage (a score out of 100) based on this 8-point total.

As a reminder, generative AI tools such as ChatGPT or Gemini are not permitted to be used when posting in the small group discussion. The response must be written entirely by the person making the contribution.

The small group discussion part of the course is worth 15% of the final grade.

Short Quiz Requirements

At various points during the course, students will be required to complete a short, 15-question objective-style quiz. The point of the quiz is to provide a "formative assessment" where both the student and the instructor can gain a sense for how well students are learning the material. Because the quizzes are formative, they only account for 10% of the final grade.

Each student will have 30 minutes to complete each quiz, and two attempts will be permitted. All of the questions will be objective-style (multiple choice or true-false), and there will only be 15 questions. No mechanism will be employed to determine if the student is using outside resources to take the quiz (e.g., the book, notes, the web, etc.); however, students are asked to take the quiz with no such resources. This is the best way for the instructor to gain a sense of the level of knowledge of the student.

Remember to answer all questions on both attempts. The system is set up to take your latest attempt, not your best attempt. It also does not average attempts.

The quizzes are designed to show all 15 questions all at once. Answers will be provided once the student submits their answers. While the time for the quiz is set for 30 minutes, it is possible to take longer. Even so, the student should strive to complete the quiz within the 30-minute timeframe.

Quiz feedback will be released automatically after the due date passes.

Grading Policy:

Grading will be based on biweekly programming assignments, small group discussions, and short quizzes. Final grades will be determined by the following weighting:

Item	% of Grade
Muddy Post Discussions (4)	15%
Small Group Discussions (4)	15%
Short Quizzes (4)	10%
Programming Projects (4)	60%

Each programming assignment will outline the specific requirements and steps to be taken to complete the assignment with associated weights. In terms of assigning a final letter grade, the following is provided as the default scheme. If deemed appropriate, the instructor may adjust these grades downward, for example, to achieve a target of 20% A's.

Score Range	Grade
[93,100]	A
[90,93)	A-
[87,90)	B+
[83,87)	B
[80,83)	B-
[70,80)	C
[0,70)	F

Students at risk of receiving a C or lower in the class are identified at midterm as part of the course roster verification. If you do not receive notice from the university or the professor following midterm, you are on track to receive at least a B- in the class.

Policies

Course Policies:

Late Submission Policy

Being that we are all working professionals, and time management is of critical importance, the purpose of this document is to lay out the course policy with respect to completing course assignments.

The policy of this course is that no late submissions will be accepted.

Note that I recognize exceptional circumstances may arise, and I am willing to work with students when they do. Therefore, the following additional requirements are put in place:

1. If you must travel for business and will have limited Internet connectivity, then you must notify the instructor at least *one week* prior to travel to make arrangements for handling assignments. Failure to provide this advanced notice will result in all due dates being enforced.
2. If you are traveling on vacation, then all due dates remain enforced. Personal travel is not an excuse to relax the due dates.
3. If there is a family emergency (e.g., a death in the family or a serious illness), then you must notify the instructor as soon as possible to make arrangements for handling the assignments. Note that this may require some form of documentation.
4. If you become personally ill, then it is important for you to take care of your health; however, since we are not meeting in person in a classroom, meaning that spread of disease is no longer an issue, only illnesses or injuries that require professional medical attention will receive special handling. Medical documentation will be required; otherwise, all due dates will be enforced.
5. Under no circumstances will time management issues result in a relaxation of due dates. Poor time management is never an acceptable excuse.
6. Special accommodations are available for students who register a disability with the university.

Questions about this policy should be directed to the instructor.

Additional Resources:

Personal Wellbeing

JHU has several resources to support students. Many students struggle with stress or a variety of mental health concerns. Student Health and Well-Being Mental Health Services has many resources available to students: MHS website: https://wellbeing.jhu.edu/MentalHealthServices/)

The Whiting School of Engineering has a dedicated office with Student Support and Advocacy. Engineering Student Support & Advocacy (ESSA) helps students navigate non-Academic issues including mental or physical health, interpersonal issues, conflict with advisors, financial concerns, time management, leaves of absence, being victimized, and family emergencies. Please find information about this office here: https://engineering.jhu.edu/studentaffairs/navigatingnonacademicissues/

MHS also partners with TimelyCare, which offers on-demand mental health support through TalkNow, as well as up to 12 free counseling appointments with the provider of your choice. Psychiatric care is also available through TimelyCare for routine medication management (no stimulants or other controlled substances). https://app.timelycare.com/auth/login.

In addition, The Johns Hopkins University Behavioral Health Crisis Support Team (BHCST) pairs experienced, compassionate crisis clinicians with specially trained public safety officers on every shift on and around the Homewood campus, seven days a week. The BHCST will provide immediate assistance to those who need it and, just as importantly, link individuals in crisis to ongoing support services in the days and weeks that follow. Call Public Safety, 410-516-5600, and ask for a BHCST clinician.

Tutoring Website

Johns Hopkins Engineering for Professionals offers a tutoring connection network that allows students to connect with other Johns Hopkins Engineering students or alumni for tutoring services. This service allows students to search a list of courses to “Find a Tutor” or complete a profile to “Become a Tutor.” More information about this service can be found on the tutoring website (https://tutor.ep.jhu.edu/).

Student Academic Success Office

If you are struggling academically, experience extenuating circumstances, or need additional academic support during the semester, the EP Student Academic Success Office (SASO) is here to help. Please email ep-studentsuccess@jhu.edu to be connected with a Student Academic Success Coordinator.

Privacy Policies:

To learn more about how to protect your data and privacy, visit Instructure's privacy policy (Canvas) and JHU's privacy policy.

Canvas Accessibility:

Online courses are taught in the Canvas learning management system. To learn more about how Canvas is designed to be accessible, visit Canvas's accessibility standards

Academic Policies:

Deadlines for Adding, Dropping and Withdrawing from Courses

Students may add a course up to one week after the start of the term for that particular course. Students may drop courses according to the drop deadlines outlined in the EP academic calendar (https://ep.jhu.edu/student-services/academic-calendar/). Between the 6th week of the class and prior to the final withdrawal deadline, a student may withdraw from a course with a W on their academic record. A record of the course will remain on the academic record with a W appearing in the grade column to indicate that the student registered and withdrew from the course.

Academic Misconduct Policy

All students are required to read, know, and comply with the Johns Hopkins University Krieger School of Arts and Sciences (KSAS) / Whiting School of Engineering (WSE) Procedures for Handling Allegations of Misconduct by Full-Time and Part-Time Graduate Students.

This policy prohibits academic misconduct, including but not limited to the following: cheating or facilitating cheating; plagiarism; reuse of assignments; unauthorized collaboration; alteration of graded assignments; and unfair competition. Course materials (old assignments, texts, or examinations, etc.) should not be shared unless authorized by the course instructor. Any questions related to this policy should be directed to EP's academic integrity officer at ep-academic-integrity@jhu.edu.

Students with Disabilities - Accommodations and Accessibility

Johns Hopkins University values diversity and inclusion. We are committed to providing welcoming, equitable, and accessible educational experiences for all students. Students with disabilities (including those with psychological conditions, medical conditions and temporary disabilities) can request accommodations for this course by providing an Accommodation Letter issued by Student Disability Services (SDS). Please request accommodations for this course as early as possible to provide time for effective communication and arrangements.

For further information or to start the process of requesting accommodations, please contact Student Disability Services at Engineering for Professionals, ep-disability-svcs@jhu.edu.

Student Conduct Code

The fundamental purpose of the JHU regulation of student conduct is to promote and to protect the health, safety, welfare, property, and rights of all members of the University community as well as to promote the orderly operation of the University and to safeguard its property and facilities. As members of the University community, students accept certain responsibilities which support the educational mission and create an environment in which all students are afforded the same opportunity to succeed academically.

For a full description of the code please visit the following website: https://studentaffairs.jhu.edu/policies-guidelines/student-code/

Classroom Climate

JHU is committed to creating a classroom environment that values the diversity of experiences and perspectives that all students bring. Everyone has the right to be treated with dignity and respect. Fostering an inclusive climate is important. Research and experience show that students who interact with peers who are different from themselves learn new things and experience tangible educational outcomes. At no time in this learning process should someone be singled out or treated unequally on the basis of any seen or unseen part of their identity.

If you have concerns in this course about harassment, discrimination, or any unequal treatment, or if you seek accommodations or resources, please reach out to the course instructor directly. Reporting will never impact your course grade. You may also share concerns with your program chair, the Assistant Dean for Diversity and Inclusion, or the Office of Institutional Equity. In handling reports, people will protect your privacy as much as possible, but faculty and staff are required to officially report information for some cases (e.g. sexual harassment).

Course Auditing

When a student enrolls in an EP course with “audit” status, the student must reach an understanding with the instructor as to what is required to earn the “audit.” If the student does not meet those expectations, the instructor must notify the EP Registration Team [EP-Registration@exchange.johnshopkins.edu] in order for the student to be retroactively dropped or withdrawn from the course (depending on when the "audit" was requested and in accordance with EP registration deadlines). All lecture content will remain accessible to auditing students, but access to all other course material is left to the discretion of the instructor.

Course Schedule

Course Schedule:

Module	Date	Module Title	Assignments
Module 1	5/28/24 – 6/3/24	Non-parametric Learning	Assigned Readings; Deliverables: Student Introduction (Day 7)

Module	Date	Module Title	Assignments
Module 2	6/4/24 – 6/10/24	Clustering	Assigned Readings; Deliverables: Small Group Discussion (Ends Day 7), Project Code (Day 7)
Module 3	6/11/24 – 6/17/24	Bayesian Learning and Decision Trees	Assigned Readings; Deliverables: Muddy Point (Day 3), Muddy Response (Day 5), Short Quiz (Day 6), Project Video and Paper (Day 7)
Module 4	6/18/24 – 6/24/24	Ensembles	Assigned Readings
Module 5	6/25/24 – 7/1/24	Dimensionality Reduction and Rule Learning	Assigned Readings; Deliverables: Small Group Discussion (Ends Day 7), Project Code (Day 7)
Module 6	7/2/24 – 7/8/24	Representation Learning and Linear Models	Assigned Readings; Deliverables: Muddy Point (Day 3), Muddy Response (Day 5), Short Quiz (Day 6), Project Video and Paper (Day 7)
Module 7	7/9/24 – 7/15/24	Feedforward Neural Networks	Assigned Readings
Module 8	7/16/24 – 7/22/24	Other Neural Networks	Assigned Readings; Deliverables: Small Group Discussion (Ends Day 7), Project Code (Day 7)
Module 9	7/23/24 – 7/29/24	Deep Learning	Assigned Readings; Deliverables: Muddy Point (Day 3), Muddy Response (Day 5), Short Quiz (Day 6), Project Video and Paper (Day 7)
Module 10	7/30/24 – 8/5/24	Reinforcement Learning	Assigned Readings

Module	Date	Module Title	Assignments
Module 11	8/6/24 – 8/12/24	Temporal Difference Learning	Assigned Readings; Deliverables: Small Group Discussion (Ends Day 7), Project Code (Day 7)
Module 12	8/14/24 – 8/19/24	Deep Reinforcement Learning	Assigned Readings; Deliverables: Muddy Point (Day 3), Muddy Response (Day 5), Short Quiz (Day 6), Project Video and Paper (Day 7)