Course Description: Machine learning is an exciting and one of the fastest growing fields of research in computer science with enormous applications ranging from biology to autonomous driving, computer vision, natural language processing, and robotics. With ever increasing volumes and ever evolving data patterns, it is evident that the need for automated data is critical. The goal of machine learning is to develop methods that can automatically detect patterns and insights within data that are interesting, informative, yet otherwise would be impossible to retrieve. Machine learning thus closely works with statistics and data mining (refresh your calculus and probability knowledge with no further delay!), but differs slightly in terms of terminologies and objectives. This course will give you an introduction to the field and briefly describe its applications in the areas of computer vision, natural language processing, time series data etc.

Course Objectives:

1. To provide a broad survey of approaches and techniques in ML
2. To understand the concept of the topics in depth along with the mathematical formulations designed to propose them.
3. To develop the design and implementation skills that will help you to build intelligent, adaptive artifacts
4. This is NOT a python programming course, we will use Python to discuss the implementation part of the concepts. However, you are free to use any other programming language if you feel at home with it. However, due to Autograding, we will have to choose a language, and that will be Python. TAs will provide you with more details on this during the course.

Course Prerequisites: Expertise in Python or atleast one Programming Language. A solid background in programming and data structures.

Requirements: The course grade will be based on \( \approx \) weekly (or biweekly) written assignments, three/four programming assignments, one/two mid-terms, and a final exam. Homework is due before class on the due date.

Grading Policy: The following items are designed to make your life easier and to give you some flexibility for planning your work:

- The lowest programming assignment and 2 lowest written assignment scores will be dropped.
• You get 7 flexible extension days for handing in assignments, with max of 2 per assignment. Use them wisely!
• Your written assignments will need to be clearly written, succinct and accurate.
• The decision to conduct the second mid-term will depend on the class performance in mid-term 1, which will be conducted during the first or second week of March.
• The Final will contain all the materials that will be taught in the course. However, material that is not covered in the two Mid-terms will have more weights.
• You will need to submit your assignments through UBLearn.
• Each programming assignment submission should carry a clearly written report with snapshots of the results (including intermediate steps). More details on the demo schedules will be updated later in the syllabus and in Piazza posts.
• Incomplete homework assignments can be turned in for partial credit. However, the reports should clearly mention about latest status of the incomplete submissions.
• The schedule mentioned below is just for your understanding of the course content, and subject to minor changes, if required.
• While attendance is not mandatory, it helps in margins. Also there will be some class activities, which will be published during the class hours and close after the class. Mostly these questions will be based on the class discussions. If you are attentive, you will also get the solutions in class. The performance in this component will add to your total as Bonus.

Grade Composition:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Assignments</td>
<td>25%</td>
</tr>
<tr>
<td>Written Assignments</td>
<td>25%</td>
</tr>
<tr>
<td>Mid-term(s):</td>
<td>20%</td>
</tr>
<tr>
<td>Final:</td>
<td>30%</td>
</tr>
<tr>
<td>Class Activities:</td>
<td>5% (Extra Credit!)</td>
</tr>
</tbody>
</table>

Make-up policy: The request should be made sufficiently in advance of the test, for valid reasons. The make-up should be scheduled before the next class.

While I will try to make the slides as detailed as possible, there is NO alternative to a book. I strongly encourage everyone to read the specific chapters in the textbooks, that will be mentioned in the class as a part of the reading material for a topic.

Textbooks/Reference Books: The softcopies of the books are available in Piazza. The reading materials for each topic will be mentioned in the class and also in the lecture slides.


Covered Topics:
Rough schedule (each topic will take up 1-3 weeks)
1. Introduction and overview (0.5 weeks).
2. Supervised Learning: Linear Models (≈ 2 weeks): This will include Linear Regression, Logistic Regression, SVM etc.
3. Generative Vs Discriminative Methods (≈ 2 weeks): Multi Layer Perceptrons / Neural Network, Understanding the Brain, Perceptron, MLP, Backpropagation, Training Stratgy, Brief introduction to DL.
4. Kernel Methods (≈ 1.5 weeks): Kernel Functions, Kernel tricks, SVM, Multiclass Kernel Learning, Using in Regression/Ranking,
5. Graphical Models (≈ 1.5-2 weeks): Chain Rule, Conditional Independence, Bayes Net, Inference, Learning.
6. Sampling Methods (≈ 1 week): Basic Sampling Methods like Rejection Sampling, Improtance Sampling, MCMC sampling, Gibbs Sampling
7. Dimension Reduction (≈ 1.5weeks): PCA, Feature Embedding, SVD, MF, LDA, CCA
8. Sequential Model (≈ 2.5 week): Markov Property, HMM, Generalization of HMM, RNN
9. Ensemble Methods (≈ 2 week)

Accessibility Resources: If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources, 25 Capen Hall, 645-2608, and also the instructor of this course .. The office will provide you with information and review appropriate arrangements for reasonable accommodations. http://www.student-affairs.buffalo.edu/ods/

Academic Integrity:
(Short) Don’t cheat! You will be caught and punished. Our department is serious about graduating ethical and upstanding computer scientists. The policy has recently been updated and will be enforced.
(Long) All academic work must be your own. Plagiarism, defined as copying or receiving materials from a source or sources and submitting this material as one’s own without acknowledging the particular debts to the source (quotations, paraphrases, basic ideas), or otherwise representing the work of another as one’s own, is never allowed. Collaboration, usually evidenced by unjustifiable similarity, is never permitted in individual assignments. Any submitted academic work may be subject to screening by software programs designed to detect evidence of plagiarism or collaboration. Also, do not post any of the course material outside of the Course piazza page. It will be interpreted as an attempt to get non-approved help. For the complete policy please see: https://engineering.buffalo.edu/computer-science-engineering/information-for-faculty-and-staff/academic-integrity.html

Approved Resources:

1. Any material posted in the slides.

2. Material from the text-book (will copy relevant content to slides). Note, the code solutions from the book’s website are NOT approved unless they are explicitly posted on the piazza page.

3. Sites (one click away) from the approved resources list on the Piazza page. I will add to them as appropriate for throughout the semester.
Working with others: Please do help each other! This material is fun, but can be challenging. Discussing it with peers can deepen your understanding. You can talk about the homework problems and ways of approaching them, however, every person must write up solutions and code separately. We will compare all submissions with each other AND non-approved sources. If you can find something online, so we can we.