International Journal on Cybernetics & Informatics (IJCI) Vol. 12, No.4, August 2023

INTEGRATING AI AND ML EDUCATION IN SMALL LIBERAL ARTS INSTITUTIONS: CURRICULUM AND PEDAGOGICAL CONSIDERATIONS

Nathan Johnson and Sayani Sarkar

Department of Computer Science, Bellarmine University, Louisville, Kentucky, USA

ABSTRACT

The proliferation of artificial intelligence (AI) and machine learning (ML) in modern institutions has necessitated integrating these technologies into academic curricula. This transition challenges small liberal arts institutions, which offer unique educational opportunities but face distinct hurdles in adapting their programs to meet the growing demand for AI and ML expertise. This article highlights the importance of providing computer science graduates from these institutions with a solid foundation in AI and ML systems. The recommended curriculum includes problem-solving techniques, algorithm design, data preprocessing, model training, and ethical considerations in AI and ML. The pedagogy emphasizes practical assignments, projects, and collaborative learning to foster critical thinking and creative problem-solving skills. Moreover, integrating AI and ML concepts across disciplines enables students to explore these technologies' broader implications and ethical dimensions. Small liberal arts institutions can leverage their unique educational environments to promote interdisciplinary collaborations and provide students with a well-rounded understanding of AI and ML applications. In conclusion, adapting small liberal arts institutions to incorporate AI and ML education is vital for preparing computer science graduates to meet the evolving demands of the modern workforce. By embracing these advancements and tailoring their programs accordingly, these institutions can equip their students with the necessary skills and knowledge to thrive in an AI-driven world.

KEYWORDS

Artificial Intelligence, Machine Learning, Small Liberal Arts Institutions, Curriculum Development, Pedagogy, Computer Science Education.

1. INTRODUCTION

The modern computer science (CS) curriculum is a busy place! Enthusiastic (we hope) undergraduate students climb a core ladder of classes, including basic programming, discrete math, data structures, and algorithms. When they graduate, they have a world of expectations, not the least of which is that they have some familiarity with artificial intelligence (AI), machine learning (ML), and data science. The market is proliferating, and one recent study indicated that the AI market will reach US$ 1,597.1 billion by 2030 (see Figure 1) [1]. In business adoption alone, worldwide spending on AI and related systems has grown during the past few years, reaching $24 billion in 2018, with more than $77 billion expected in 2022 [2].

Of course, college and university CS programs often include electives such as AI, machine learning, data science, robotics, and related topics, such as cybersecurity or cloud computing. Students advance through the curriculum, choosing from the entire array of electives every term, getting into each class when they choose and when needed. Well, not so much! Smaller schools, particularly liberal arts colleges, and universities, don't have the faculty to offer all

Dhinaharan Nagamalai (Eds): EDU, SCMC -2023
pp. 01-17, 2023. IJCI – 2023 DOI:10.5121/ijci.2023.120401

electives each term. In addition, the student population isn't sufficient for multiple electives to have enough course enrollment for the course to be delivered. Moreover, degree requirements limit the number of electives students are allowed to take. So many students may graduate without experience in some topics – usually, a venial sin so long as the core is maintained. However, we would argue that two subjects, AI and ML, are fundamental in a modern CS program. New CS graduates almost certainly will find themselves, at least tangentially, involved with some work-related aspect of AI, ML, data science, or ancillary area. AI is evolving into a critical study area as it infiltrates various business and government information systems.

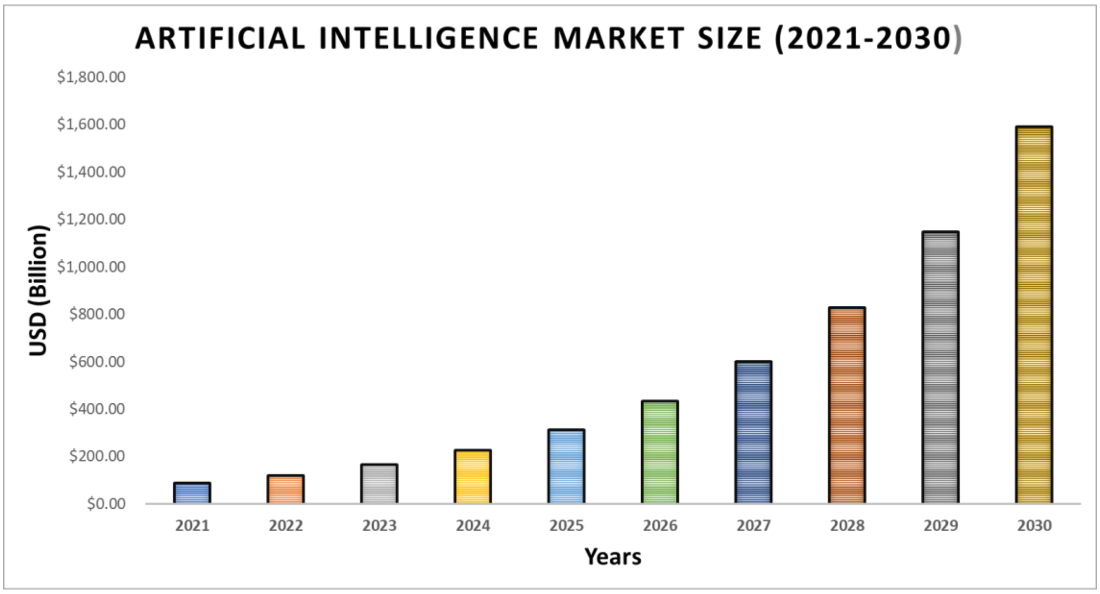

Figure 1. Exponential Growth of the Artificial Intelligence Market

At many schools, classes such as AI, ML, and data science are offered as electives intermittently, not as part of the core. Other areas compete for attention, including software engineering, databases, networks, compilers, graphics, etc. It's a long list. Some students might find themselves in some AI course – if they have time and for small schools at least, if the timing is right, or they might find themselves in an ML course or a course that involves data models. Rarely will they be able to take all three, at least not at smaller schools with limited faculty.

However, many schools do not include ML in that core set of classes. Shapiro, Fiebrink, and Norvig address the issue in their 2018 paper: "The growing importance of ML thus raises challenging questions for CS education: How should practical and theoretical ML topics now be integrated into undergraduate curricula? Furthermore, how can we make room for expanded ML content in a way that augments— rather than displaces—classical CS skills within undergraduate degree programs whose duration must remain static? " [3]

Additionally, a 2020 publication from a task force including representatives of the ACM and IEEE, among several other industry and academic stakeholders, points out that AI and related areas of computing "have blossomed during the last ten years. AI and its allied field of robotics have become prevalent fields of study in computing. Although no formal professionally endorsed AI curriculum exists at the time of this writing, a curricular recommendation in these areas has the potential to emerge in the next few years " [4].

Looming in the not-to-distant future is another challenge for traditional CS education – AI systems that themselves can write and explain computer code, among other things. ChatGPT [5] recently debuted on the internet and can quickly write and debug complicated computer code successfully. In any case, there are two questions at work here. First, why should small liberal arts institutions concern themselves with computer science? Why not leave it to large engineering programs? Secondly, given the small student body and faculty and staff limitations, it is possible to provide students with a grounding in AI and ML while maintaining the necessary core classes for a CS degree.

In this article, we will briefly consider the unique role of smaller liberal arts institutions in CS education and the limitations such institutions face in adding AI topics to the current curriculum. Finally, we examine how smaller programs might introduce CS majors to the basics of AI, data modeling, and ML in a single, one-term course. We will offer as an example the AI course we provide at Bellarmine University in Louisville, KY. Bellarmine University has an undergraduate enrollment of 2,484 students as of 2020 and a graduate enrollment of 809 students.

2. LIBERAL ARTS AND COMPUTER SCIENCE EDUCATION

Traditionally large CS and engineering programs have concentrated on providing technical and engineering expertise for CS majors. Any additional courses usually focus on math, with the remaining hours filled by general education requirements. Liberal arts education includes another mandate that provides for but differs from general education, what can be thought of as the core part of the curriculum, in that it should be intentionally interdisciplinary and reflective, requiring "the multitude of perspectives, ways of thinking, methods and knowledge content anchored in a variety of disciplines. It requires its students to study beyond a single subject or within one family of disciplines… it lays the foundation for learning how to interpret, interrogate or to make new knowledge framed in the constructs of various fields " [6]. As Wolfson, Cuba, and Day point out, the benefits of liberal arts education seem to accrue most to those who delve deeply into a second field rather than surveying a variety of areas of study and "general education" coursework [7]. As a surprising fact, liberal arts colleges produce proportionally more PhDs in science [8].

The multidisciplinary approach to CS that a liberal arts institution can provide is precious to those entering the field. It is important to note that CS now plays a role in almost every study area. CS is everywhere. All science is computer science [9]. In other words, CS has become central to virtually all fields of study in some fashion and could be called the most liberal of the liberal arts. As Steve Jobs, former Apple CEO, put it: "… science and CS is a liberal art; everyone should know how to use, at least, and harness in their life. It should not be relegated to 5 percent of the population over the corner. It's something that everybody should be exposed to, and everyone should have mastery of to some extent, and that's how we viewed computation and these computation devices" [10].

In their 2022 paper, Brodley et al. describe their program at Northeastern University, where students "can choose among three computing majors like CS, Data Science or Cybersecurity and 42 combined majors, which combine of the three computing degrees with one of 29 distinct majors in other fields" [11]. One effect of the program has been to create greater gender diversity in CS majors. The term "X+CS" is a similar concept whereby students in fields of study other than CS add CS to their major. Liberal arts institutions are in a position not only to combine specific programs to form majors but to allow students to mix and match classes, creating their degree programs.

Smaller schools have advantages and limitations. Smaller enrollment means classes often are taught by faculty with terminal degrees and excellent student-teacher ratios, with the result of more personal attention. The core philosophy of liberal education focuses on analytic thinking and social responsibility. However, liberal arts colleges and universities are constrained in many ways that do not face larger institutions. They face a mandate to graduate students with a wider variety of educational experiences. Smaller enrollment means class sizes tend to be smaller, and a crisis in student demographics is likely to make enrollments smaller or constrain their growth[12]. Some classes will not have enough students to be offered every term or year. This burdens students to show at least intermediate competence in the second field of study, limiting the time and course availability for their major. This brings us back to AI and ML. With all of the constraints we have placed on liberal arts students in CS programs, there is room for additional classes – data science, AI, and ML. Students might have one opportunity to take one course in one of those topics in their four-year tenure or find themselves with only one. At the very least, they must be introduced to those topics before leaving university.

2.1. Skills Requirements for Computer Science + X Graduates

The field of CS is rapidly evolving, and the demand for professionals with interdisciplinary skills is on the rise. This abstract explores the skills requirements for CS + X graduates, where X represents a complementary discipline or specialization. CS + X graduates are expected to possess a strong foundation in CS principles, including programming, data structures, algorithms, and software development. Additionally, they need to acquire domain-specific knowledge in their chosen X field, ranging from biology and finance to psychology and design.

Furthermore, interdisciplinary graduates should possess excellent problem-solving and critical-thinking abilities. They should be able to analyze complex challenges and apply their CS and X knowledge to develop innovative solutions. Communication and collaboration skills are crucial, as graduates often work in multidisciplinary teams and must effectively convey technical concepts to non-technical stakeholders. AI skills are near the top of the list shown in Table 1 [13, 14]. As Loureiro et al. put it in their wide-ranging survey paper, "To encourage further advancements in research on business applications of AI, which often require a multidisciplinary perspective, AI practitioners and researchers will benefit from a comprehensive knowledge about what has been investigated and applied in different business domains (i.e., from manufacturing to services) and in different disciplinary fields, such as marketing, tourism, management, sociology, psychology, and so on. Such comprehensive knowledge will provide researchers with a foundation to prioritize research foci and practitioners to guide effective investment in important aspects of AI for business."

Table 1. Top 5 specialty and skills for CS+X graduates

Sl No.	**Tech-focused Specialty**	**Skills**
1	Software Development	Build things that function in the real world. The programming languages are Java, .NET/C#, JavaScript, or Python
2	AI	Programming Skills, Libraries, and Frameworks, Mathematics and Statistics, Machine learning, deep learning, and natural language processing
3	InfoSec And Cybersecurity	Enterprise privacy and security and making sure their data is safe from attacks from both internal and external bad actors
4	Big Data	Hands-on experience in Java/ C++/ Python. Database and SQL: In-depth knowledge of DBMS and SQL
5	Data Analytics	Data cleaning and preparation, data analysis and exploration, statistical knowledge, data visualizations, problem-solving

We agree! Broad education and interdisciplinary thinking are at the heart of liberal arts education. We would argue that the same comprehensive knowledge is at least as valuable for any CS graduate. Put the other way, recent CS graduates need a broad introduction to many fields, especially if they are working in an area of ML or AI.

Good sense and avoiding an existential crisis may require that small liberal arts institutions cleave to their purpose and design CS programs that leave plenty of room for other subjects, both in science and the humanities [15]. CS majors at small liberal arts institutions should be able to graduate with a minor, second major, or combined program in practically any other subject they choose. Graduates from CS programs at liberal arts institutions should be able to bring technical/theoretic skills in CS, communication, and social analytics, reasoning skills, broad knowledge of various subject areas, and reasonable depth of knowledge in at least one area besides CS.

3. Artificial Intelligence and the Computer Science Curriculum

We have discussed the benefits of a multidisciplinary and pedagogically liberal approach to teaching CS and the demand for some knowledge of AI and ML. At this point in our journey, we have argued that CS students at small liberal arts institutions will be exposed to liberal curricula combining and fostering introspection and reflection in various disciplines. However, where does that leave us regarding the core CS curriculum? In addition to serving many other fields of endeavor, for better or worse, CS keeps broadening, adding subfields, from cybersecurity to cloud systems to data science and ML. Some, such as ML, differ significantly from the traditional CS curriculum. What are some of the top technical skills required in today's workplace? The skills domain includes CS, analytics, computer engineering, and security. Software development and AI are the top two in-demand skills for CS graduates, including one or more programming languages like JAVA, C++, and Python. For AI programming, understand mathematics, statistics, and ML algorithms. The workplace is demanding, however, and many other desired skills exist. Well-rounded graduates in CS should have some knowledge of many or most of these areas.

3.1. Qualifications for AI/ML Job Positions in the Current Job Market

Indeed, the demand for AI, data science, and ML skills is increasing across various industries, including business, government, scientific, medical, and military sectors. However, it can be challenging for CS students to dedicate ample time to specialized courses focused on AI, ML, or data science due to the breadth of topics in their degree programs.

The qualifications you listed from job descriptions on leading recruiting sites, specifically from Google [16], emphasize the following skills and requirements:

1. Education: Bachelor's degree in CS, engineering, math, physics, or a related discipline with coursework in AI/ML.
2. Programming Skills: Experience in software development using Python and C++.
3. ML Fundamentals: Understanding of ML and deep learning fundamentals.
4. AI/ML Technologies: Experience with AI, ML, and RL learning technologies.
5. Tooling and Deployment:
    - Ability to develop internal tooling to support ML efforts.
    - Production deployment and performance monitoring, including code refactoring and optimization as necessary.

6. Programming Languages and Software Development:
   - Experience in Python and R programming.
   - Understanding of the software development life cycle.

While the job descriptions highlight these qualifications, it is worth noting that different companies may have varying requirements and priorities based on their specific needs and projects. However, a strong foundation in CS, programming skills in Python and C++, and a fundamental understanding of ML concepts are commonly sought after.

Students interested in AI and ML must supplement their education with self-study, online courses, or specialized programs to gain practical experience and in-depth knowledge. Additionally, hands-on projects, internships, or research opportunities can help students develop the skills required for AI and ML careers.

3.2. Liberal Arts Institutions Offering AI in their Computer Science Curriculum

Numerous liberal arts colleges and universities have recognized the growing significance of AI and ML in today's technological landscape. These institutions, known for emphasizing a well-rounded education and interdisciplinary approach, are integrating AI and ML topics into their CS programs. By combining the principles of CS with the broader perspectives offered by a liberal arts education, these institutions are preparing their students for the rapidly evolving field of AI. Here is a list of liberal arts institutions shown in Table 2 with curriculum highlights in CS programs, specifically in AI/ML and data science:

Table 2. Selected CS Programs Offering AI and ML Courses

Institution (reference)	Course Name	AI Curriculum Highlights
Carleton College [17]	CS 321: Making Decisions with AI	▪ Intelligent search strategies ▪ Game-playing approaches, ▪ Constrained decision making ▪ Reinforcement of learning from experience
Swarthmore College [18]	CPSC 063: AI	▪ Neural networks ▪ Decision trees ▪ Genetic algorithms ▪ Reinforcement techniques
Williams College [19]	CSCI 373: AI	▪ Problem-solving by search ▪ Logic and Planning ▪ Constraint satisfaction problems ▪ Reasoning under uncertainty ▪ Probabilistic graphical models ▪ Automated Learning
Wellesley College[20]	CS 232: AI	▪ Symbolic AI, ▪ Rule-based systems ▪ Statistical approaches that rely on increasingly large amounts of data ▪ Contemporary deep-learning techniques
Amherst College [21]	COSC 241: AI	▪ Adversarial game playing ▪ Heuristic search ▪ Design of agents that learn either from experience or from a provided dataset
Davidson College [22]	CSC 370: AI	▪ Search, game playing, constraint satisfaction problems, planning ▪ Reinforcement learning ▪ Knowledge representation and logic.
Reed College [23]	CS 377: AI	▪ Knowledge representation ▪ Reasoning under uncertainty ▪ Logic programming, planning ▪ Algorithmic strategies for large-scale combinatorial search
Colorado College [24]	CS 3XX: AI	▪ Problem-solving and game playing ▪ Knowledge representation ▪ Natural language understanding ▪ Expert systems

4. Computer Science Programs at Bellarmine University: BA Degree with Artificial Intelligence & Machine Learning Course

Bellarmine University offers a Bachelor of Arts (BA) degree in CS that provides students with a comprehensive education in CS while incorporating additional interdisciplinary elements. This program covers the core concepts of CS and emphasizes the integration of mathematics, physics, foreign language proficiency, and the flexibility to pursue a variety of minors or second majors. Notably, the program recently introduced a new AI class encompassing classical AI, data science, and ML in a single term.

The BA degree program at Bellarmine University mirrors the structure of a traditional Bachelor of Science (BS) degree in CS by including a mathematics minor and university-level physics courses. This rigorous foundation ensures that students gain strong analytical and problem-solving skills essential for success in CS. By incorporating these mathematical and scientific components, the program prepares students for the technical challenges they may encounter in their careers.

However, recognizing the evolving landscape of CS and students' diverse interests, Bellarmine University has also introduced a newly designed BA degree option. This alternative BA program reduces the emphasis on mathematics and allows students to pursue a minor or second major of their choice. This modification provides students with increased flexibility in tailoring their degrees to align with their interests and career goals.

Bellarmine University has developed an AI class as part of its CS curriculum in line with the growing importance of AI and ML. This course offers students a broad overview of classical AI, data science, and ML within a single term. By covering these fundamental areas, students gain exposure to the theoretical concepts, practical techniques, and ethical considerations related to AI and ML. This comprehensive approach allows students to understand the interplay between different aspects of AI and develop a well-rounded perspective on the field shown in Table 3.

Table 3. Proposed Bachelor of Arts in Computer Science Course List

Year	Course	Course Description
Freshman (1st year)	CS 1XX: Programming Fundamentals:	▪ Introduction to fundamental concepts of procedural programming ▪ Data types, control structures, functions, arrays, and files ▪ The mechanics of running, testing, and debugging; problem-solving techniques
	CS 1XX: The Object-Oriented Paradigm	▪ Introduction to the concepts of object-oriented programming ▪ Definition and use of classes along with the fundamentals of object-oriented design ▪ Inheritance and polymorphism ▪ A simple analysis of algorithms ▪ Basic search and sorting techniques and an introduction to software engineering issues ▪ Introduction to generic programming
Sophomore (2nd year)	CS 2XX: Data Structures:	▪ Algorithmic notation ▪ Algorithm design ▪ Elementary data structures and their storage representations ▪ Linear data structures and their sequential and linked representations ▪ Nonlinear data structures and their storage representations ▪ Memory management, file processing ▪ Sorting and searching algorithms
	CS 2XX: Logic Design	▪ Introduction logic gates ▪ Combinational and sequential circuits ▪ Circuit simplification using Karnaugh maps and Boolean functions ▪ Flip-flops as employed in semiconductor memories ▪ Counters and registers ▪ Electronic implementation of binary arithmetic

Junior (3rd year)	CS 3XX: Operating Systems	▪ Classification schemes for operating systems ▪ Resource-manager model of an operating system, system structure ▪ Memory management, process management, design techniques ▪ Implementation of a simple operating system and related software
	CS Elective I	▪ Listed below
	CS 3XX: Compiler Construction	▪ The purpose of compilers ▪ Different types of compilers ▪ Formal language concepts, including syntax and essential characteristics of grammar ▪ Linguistic analysis and parsing techniques ▪ Interpretative languages, assembly language
	CS 3XX: Algorithms	▪ Algorithm design techniques, including backtracking, heuristics, recursion, and simulation ▪ Experimental and analytical determination of algorithm performance Applying algorithm design to various areas of CS, such as AI and systems programming.
Senior (4th year)	CS 4XX: Overview of AI	▪ As discussed in Table 4
	CS 4XX: Software Design and Development	▪ Design techniques, formal models of structured programming, organization, and management ▪ Estimating program libraries, documentation, and organization of a large-scale project by students.
	CS Elective II	▪ Listed below
	CS 4XX: Data Communications and Computer Networks	▪ Traditional star networks vs. various distributed designs; access methods and protocols ▪ Data communications hardware ▪ Software and transmission media ▪ Systems design considerations ▪ Implementation and upgrading of computer networks.
	CS 4XX: Computer Science Capstone	▪ Satisfactory completion of a significant design and development project with a written report and an oral presentation is required. ▪ Includes a comprehensive exam in CS as appropriate
CS Electives	▪ Aerial Robotics ▪ Database management system ▪ Machine learning on Cloud ▪ Visual Programming	

CS students might have the opportunity to take, at most, a single term in AI. At many institutions, these introductory courses cover one aspect of AI. Such a first course in AI might cover "classic" symbolic AI along with heuristic search strategies and game theory followed by predicate logic and so on. Alternatively, a first course might delve exclusively into various ML/data science models with a nod toward neural networks.

However, it is rapidly becoming incumbent upon educators to retain important aspects of symbolic AI and an overall paradigm for thinking about AI while giving students some experience and familiarity with ML models and neural networks in a practical way -- and somehow do it in a single, one-semester course. It is a nuanced and overwhelming problem.

5. A PROPOSED FIRST COURSE IN ARTIFICIAL INTELLIGENCE

To effectively address the requirements of CS students in today's AI-centric work environment, we propose a comprehensive first course in AI that covers essential theoretical foundations, classic AI algorithms, ML concepts, and basic neural networks. While time constraints limit the breadth of topics that can be covered, this course aims to provide a practical and broad basis for students to grasp AI concepts quickly in future academic or professional settings.

The course outline includes the following key components:

1. Establishing a Theoretical Foundation:
   - Introduce students to the fundamental principles and concepts that underpin the field of AI.
   - Cover intelligent agents, problem-solving, knowledge representation, and reasoning topics.
2. Classic AI Algorithms:
   - Familiarize students with commonly used and well-established algorithms in classic AI.
   - Provide basic programming and analysis skills to implement and understand these algorithms.
   - Explore topics like search algorithms, constraint satisfaction problems, and game playing.
3. Introduction to ML and Data Science:
   - Introduce students to essential concepts in ML and data science.
   - Emphasize a hands-on and programmatic approach to understanding these concepts.
   - Cover supervised and unsupervised learning, regression, classification, and evaluation metrics.
4. Connectionist Concepts and Basic Neural Networks:
   - Familiarize students with connectionist models and basic neural networks.
   - Introduce the workings of neural networks and their applications in AI.
   - Cover topics like perceptron, activation functions, feedforward networks, and backpropagation.

While it is not feasible to cover every topic in classic AI, data science, and ML within the limited timeframe of the course, careful selection of individual topics ensures that students gain a broad and practical understanding. Students are assumed to have already taken prerequisite courses such as data structures, algorithms, and discrete mathematics. Some topics, such as predicate logic, may have been adequately covered in discrete math and can be referenced as needed in the AI course.

Specific topics are deemed indispensable for problem-solving in AI and should not be omitted. For example, various graph search strategies serve as essential techniques for problem-solving in AI and general agent-based problem-solving scenarios. By including such material, the course equips students with the necessary skills for AI problem-solving and provides a foundation for future learning in the field.

Overall, this proposed first course in AI aims to balance theoretical foundations, practical implementation, and exposure to crucial AI subfields. By carefully selecting and covering essential topics, students are provided with a solid basis to further exploration in subsequent courses, research endeavors, or professional AI work. Many topics are covered in our proposed course (see Figure 2).

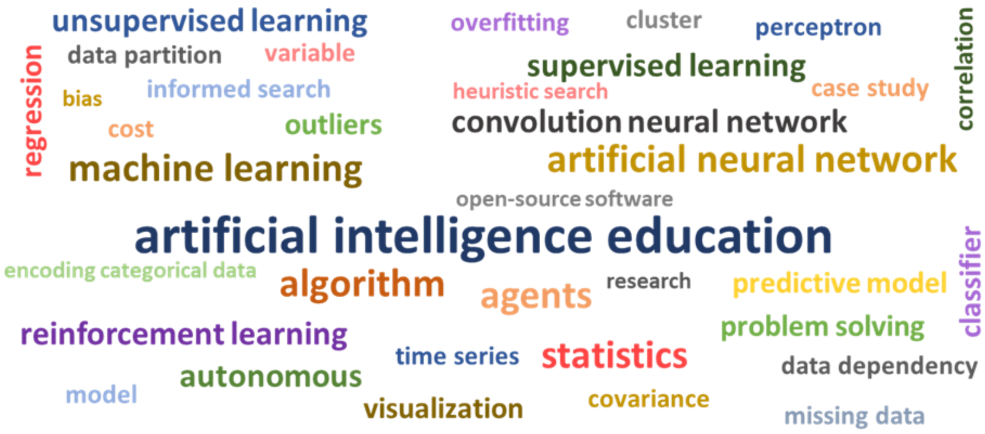

Figure 2. Topics in a proposed AI course

The proposed course content and pedagogy attempt to answer many requirements for CS students facing AI in the current and near-future work environment. The course contains two major sections. The first part covers the concept of a "rational agent" operating in an environment using "percepts" and "actuators," along with theoretical knowledge of general problem-solving using traditional AI techniques [25]. The second part consists of various ML models for real-world problem solving, including supervised and unsupervised techniques, followed by an introduction to neural networks.

The course at our institution uses arguably the most popular AI text, "Russell and Norvig's AI: A Modern Approach," before moving away from the text in the second half of the course. Russell and Norvig are used to establish a paradigm or framework for considering the problem of AI in general. The second half of the term is used to discuss ML. It begins with a discussion of supervised and unsupervised learning, followed by a meeting of the theoretical foundation of linear regression with examples with Python.

Various resources are used to teach the data model-centric portion of the course, but no actual text is required. Part 1 discusses symbolic AI, and Part 2 introduces ML. The topics, instructional materials, and learning activities are summarized in Table 4.

Table 4. The First Course on Artificial Intelligence Topics

Topics	**Instructional Materials**	**Learning Activities**
1.0 Introduction to Artificial Intelligence	▪ Definition of AI ▪ Fields that contribute to AI ▪ Historical milestones that have led to our present view of AI ▪ The current state of AI	Syllabus Quiz: Understand the syllabus Assignment: P vs. NP and summary paper
1.1 Intelligent Agents	▪ The concept and types of Rational Agents ▪ Percepts and actuators ▪ Various types of environments ▪ Performance measures	Assignment: Code a simple reflex agent vacuum world with performance measures
1.2 Solving Problems by Searching	▪ Types and cost of uninformed searches ▪ Search algorithm step by step ▪ Search to find a goal state ▪ State Spaces	
1.3 Informed Search	▪ Types and cost of informed searches ▪ Various informed search algorithms ▪ How search is used to find an optimal goal state ▪ Heuristics search	Team Assignment: Using a map of Romania (from Norvig), draw search trees/stacks/queues for Depth First Search, Uniform Cost Search, Greedy Best First Search, A* search
1.4 Game Theory and Adversarial Search	▪ Game theory background ▪ Minimax algorithm ▪ Game Tree Pruning	Assigns a graph. Students make a video showing graph traversal using Alpha-Beta Pruning.
2.0 ML Overview and Definition	▪ Supervised Learning ▪ Unsupervised Learning ▪ Reinforcement Learning	Midterm Exam over Part 1 of the course
2.1 Data Preprocessing and Introduction to Anaconda Python	▪ Python Libraries, Anaconda, Jupyter Notebook ▪ Missing Data and Outliers ▪ Encoding categorical variable ▪ Data Partitioning ▪ Data Dependency ▪ Overfitting	
2.2 Introduction to Linear Regression	▪ Linear Regression • Data Visualization and Dimension Reduction • Correlation • Covariance ▪ Explanation vs. Prediction vs.ML	
2.2 Linear Regression with the Boston Housing Dataset	▪ Preprocessing Boston Housing with Python ▪ Training and validation Sets ▪ Simple regression on the dataset ▪ Correlation: Pearson Coefficient ▪ Interpreting of Least Squares	Linear Regression Lab (See Table 5)
2.3 Multiple Linear Regression	▪ Simple vs. Multiple Linear Regression ▪ P-value, R, R-Squared and adjusted R-Square, RSME, and other	Project 1 Proposal and ML project. Students select a dataset, use linear regression to model it and present their

	essential statistics ▪ Variable Reduction ▪ Correlation Matrix Python ▪ Building the regression model ▪ Model Results ▪ Feature Selection, stepwise Refinement, best Subsets	material in video and class presentation format.
2.4 Introduction to Artificial Neural Networks	▪ Perceptron ▪ Weights/Thresholds/bias ▪ Linear Regression with a Perceptron ▪ Sigmoid Functions and Sigmoid Neurons ▪ Recurrent Neural Networks vs. Convolution Neural Networks ▪ Deep neural network on Mnist dataset ▪ Gradient Descent, cost Function ▪ Back Propagation ▪ Introduction to the google cloud platform	
Final Project on ML topics (Group of two to three)	A video of your PowerPoint presentation and an explanation of your running code. ▪ KNN (K-nearest neighbors) ▪ Naïve Bayes ▪ Classification and Regression Trees (CART) (Decision Trees) ▪ Logistic Regression ▪ Association Rules (Apriori, Market Basket) ▪ Cluster Analysis	

Table 4 presents an outline of the course, highlighting the initial seven weeks that delve into the history of AI and its subfields. The course adopts the rational agent concept by Russell and Norvig, which emphasizes the importance of context and ontology in AI education. This concept views various applications, such as pricing homes in a changing economic landscape, pathfinding drones, or even hypothetical galactic overlords, as rational agents perceiving their environment through sensors and acting through actuators.

According to Russell and Norvig, a rational agent should select actions based on percept sequences and built-in knowledge to maximize its performance measure. The agent function maps percept sequences to actions and can range from simple strategies like A* search to complex neural networks that learn as they interact with the environment. This framework provides a contextual foundation for studying AI techniques, encompassing search strategies and ML methods.

The first half of the course covers essential topics, including approaches to AI, rationality, intelligence, and AI contributions from other disciplines, Alan Turing's definition of intelligence, and the Turing Test. Additionally, students gain an understanding of critical concepts such as agents, rational agents, percepts, agent functions, PEAS, observable environments, single and multi-agent systems, deterministic and stochastic systems, episodic and sequential systems, static and dynamic systems, discrete and continuous systems, and various types of agents.

The subsequent module focuses on space searches, progressing from basic tree searches to uniform cost and greedy searches and culminating in the A* search. The final module in this course part explores the game theory and the minimax algorithm, including alpha-beta pruning.

The second half of the course centers on ML, where students delve into different data models within the context of a learning agent. The curriculum commences with lectures covering supervised learning, unsupervised learning, reinforcement learning, and pertinent topics like Python Libraries, Anaconda, Jupyter Notebooks, Missing Data, Outliers, Encoding Categorical Variables, Data Partitioning, Data Dependency, and Overfitting.

Following this, classroom sessions blend lectures and hands-on lab work, emphasizing student engagement during class periods. The remaining coursework encompasses several machine-learning models and neural networks. Students are initially introduced to an overview of learning, including supervised and unsupervised learning, along with other topics outlined in Table 4. Subsequently, students follow the instructor's guidance as they progress through a linear regression example using the Boston Housing dataset. They preprocess and partition the dataset and then implement the model in Jupyter Notebooks during multiple class sessions using their laptops, covering the material outlined in Table 5.

Table 5 . Linear Regression lab.

• Preprocessing the Boston Housing dataset: Students will import the dataset and perform necessary preprocessing steps, such as handling missing data, encoding categorical variables, and scaling numeric features. • Splitting the dataset: Students will split the dataset into training and testing sets to evaluate the performance of the regression model. • Building a linear regression model: Using the training set, students will build a linear regression model using the scikit-learn library. They will fit the model to the training data and examine the learned coefficients. • Evaluating the model: Students will use the trained model to make predictions on the testing set and evaluate its performance. They will calculate metrics such as mean squared error (MSE) and R-squared to assess the model's accuracy. • Visualizing the results: Students will create visualizations, such as scatter plots and regression lines, to understand the relationship between the independent and target variables. • Interpreting the model: Students will interpret the coefficients of the linear regression model to understand the impact of each feature on the target variable.

After completing the linear regression example with the Boston Housing dataset, students are encouraged to further apply their knowledge by selecting another dataset and developing a linear regression model. However, this project alone does not cover all the necessary material. A second-class project is introduced to ensure comprehensive learning, where students engage in small-group research on various ML models.

The course includes several ML models for exploration, namely K-nearest neighbors (KNN), Naïve Bayes, Classification and Regression Trees (CART) (Decision Trees), Logistic Regression, Association Rules (Apriori, Market Basket), and Cluster Analysis. To efficiently cover the material and provide experience in dataset selection and research, students are encouraged to self-organize into groups of two or three. Each group is assigned a specific learning model from the list and provided with a compilation of familiar internet data sources. They propose an ML project utilizing the assigned learning model.

Towards the end of the course, students present their findings to the rest of the class in a 20-minute presentation. During these presentations, they explain their chosen ML model, introduce the dataset they worked with, discuss the Python code they employed, and showcase their results. Essentially, they take on the role of teaching the rest of the class about their chosen learning model.

Furthermore, the course covers relevant topics through lecture material to ensure students have a fundamental understanding of connectionist theory, neural networks, and ML. The material about neural networks includes concepts such as the Perceptron, weights, thresholds, and bias, linear regression with a perceptron, sigmoid functions and sigmoid neurons, recurrent neural networks versus convolution neural networks, deep neural networks applied to the MNIST dataset, gradient descent, cost functions, and backpropagation.

By incorporating these elements into the course, students gain practical experience in implementing ML models, conducting research, and presenting their findings to their peers. They also build a solid foundation in neural networks and ML theory.

6. SUMMARY

Institutions are spending billions on AI systems that include ML and data science, an amount that is only likely to grow in the future. Most current CS students are well-grounded in traditional procedural programming techniques, data structures, algorithms, and associated materials. Students at smaller schools may or may not be exposed to more current material involving ML models and neural networks, which could be a detriment to them as they pursue their careers. Students in a liberal arts institution face additional challenges as they try to gain and integrate a broad knowledge base with their CS major or minor. The institutions often have to navigate a narrow set of resources as they select the courses they will include in the curriculum.

We have shown one way that a great variety of important material can be conveyed to students in a single AI class. First, introducing them to classical paradigms in AI and then moving into ML models such as linear regression, CART, association rules, and others. Then we examine neural network theory and discuss CNNs and RNNs.

Finally, we discuss a pedagogy that managed to cover this material in a single course at Bellarmine University in Louisville, KY. More research is needed to examine the outcome and capabilities of our students regarding their work experience in AI and ML. In the meantime, however, we must do our best to provide students with the appropriate background for their challenges. Liberal arts education allows students to integrate diverse areas of interest, including the humanities, math, and the sciences producing not only Ph. D.s but leaders in many areas of our society and culture, a society that, without a doubt, will be dramatically changed by AI and ML.

REFERENCES

[1] Precedence Research, (2022) Artificial intelligence market. https://www.precedenceresearch.com/artificial-intelligence-market

[2] Loureiro, S. M. C., Guerreiro, J., & Tussyadiah, I. (2021). Artificial intelligence in business: state of the art and future research agenda. Journal of business research, 129, 911-926.

[3] Shapiro, R. Benjamin, et al. "How machine learning impacts the undergraduate computing curriculum." Communications of the ACM 61 (2018): 27 - 29.

[4] Force, C. T. Computing Curricula (2020). Paradigms for Global Computing Education November 2020.

[5] ChatGPT. OpenAI. https://openai.com/.

[6] Godwin, K. A., & Altbach, P. G. (2016). A historical and global perspective on liberal arts education: What was, what is, and what will be. International Journal of Chinese Education, 5(1), 5-22.

[7] Wolfson, A. J., Cuba, L., & Day, A. (2015). The liberal education of STEM majors. Change: The Magazine of Higher Learning, 47(2), 44-51.

[8] NSF (2008) https://www.nsf.gov/statistics/infbrief/nsf13323/

[9] Johnson, G. (2001). All science is computer science. The New York Times, 25, 1-5.

[10] Steve Jobs (2011). Computer Science Is a Liberal Art', Fresh Air NPR

[11] Brodley, C. E., Hescott, B. J., Biron, J., Ressing, A., Peiken, M., Maravetz, S., & Mislove, A. (2022, February). Broadening Participation in Computing via Ubiquitous Combined Majors (CS+ X). In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (pp. 544–550).

[12] Boeckenstedt, Jon (2022). Will Your College Survive the Demographic Cliff? Chronicle of higher education, volume 68, Issue 15, (2022, April).

[13] Indeed, (2021). 51 In-Demand Tech Skills for Technology Careers. https://www.indeed.com/career-advice/career-development/in-demand-tech-skills

[14] Helman, (2022). 15 In-Demand Tech-Focused (And Tech-Adjacent) Skills And Specialties. https://www.forbes.com/sites/forbestechcouncil/2022/07/11/15-in-demand-tech-focused-and-tech-adjacent-skills-and-specialties/?sh=4bdd92e138d8

[15] Campion, L. L. (2022). Leading Through the Enrollment Cliff of 2026 (Part II). TechTrends, 66(1), 112-113.

[16] Google, (n.d.). Artificial intelligence job search. https://www.google.com/search?q=artificial+intelligence+jobs+search

[17] Carlton, (n.d.). Making Decisions with Artificial Intelligence. https://www.carleton.edu/computer-science/courses/

[18] Swarthmore, (n.d.). Artificial Intelligence. https://catalog.swarthmore.edu/preview_course_nopop.php?catoid=7&coid=7880

[19] Williams, (n.d.). Artificial Intelligence. https://csci.williams.edu/courses/

[20] Wellesley, (n.d.). Artificial Intelligence. https://catalog.wellesley.edu/courses.php?pos=25&doc_type=cs232

[21] Amherst, (n.d.). Artificial Intelligence. https://www.amherst.edu/academiclife/departments/courses/1112F/COSC/COSC-241-1112F

[22] Davidson, (n.d.). Artificial Intelligence. http://catalog.davidson.edu/preview_course_nopop.php?catoid=15&coid=18641

[23] Reed, (n.d.). Artificial Intelligence. https://www.reed.edu/catalog/programs/dept_majors/csci.html

[24] Colorado, (n.d.). Artificial Intelligence. https://www.coloradocollege.edu/academics/curriculum/catalog/courses/cp365.html

[25] Norvig, Russell, (2020). Artificial intelligence is a modern approach. Pearson Education, Inc.

AUTHORS

Dr. Nathan P. Johnson joined the faculty at Bellarmine University's expanding computer science department in the fall of 2020, bringing a varied background that includes academic and work experience in technology and communications. Dr. Johnson worked for many years at the University of Louisville's Speed School of Engineering, helping researchers write parallel programs and administering the university's first "supercomputer," an IBM RS600 that was 235th on the top 500 list world's fastest computers.

Dr. Sayani Sarkar joined Bellarmine University's Computer Science area in the fall of 2021. She earned her Ph.D. in Computer Engineering from the School of Computing and Informatics at the University of Louisiana at Lafayette in 2021. For her dissertation research, she investigated intelligent energy-efficient drones in path planning, monitoring, and real-time decision-making. Dr. Sarkar's current research interests include intelligent UAV path design, modeling, and analysis using deep learning algorithms.