ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022

Paper ID #37675

Work-in-Progress: Development of an Introductory Machine Learning Course in Biomedical Engineering

Patjanaporn Chalacheva (Assistant Teaching Professor)

Work-in-Progress: Development of an Introductory Machine Learning Course in Biomedical Engineering

Introduction: Artificial intelligence and machine learning (AI/ML) play an increasingly important role in driving new technological developments in all aspects of life. With growing interest in AI/ML among biomedical engineering students, the introductory course to machine learning offers students an opportunity to grow their expertise in this new area and to learn how AI/ML is used in biomedical engineering applications. This work outlines the development of an introductory machine learning course for biomedical engineering (BME) graduate and advanced undergraduate students, and discusses the topic covered, challenges encountered in the early iteration of the course (Fall 2020), and how may these challenges be addressed in the future.

Methods: The Introduction to Machine Learning for Biomedical Engineering course introduces high-level concepts behind machine learning (ML) algorithms and teaches students which ML algorithms are best suited to different kinds of biomedical-related problems. The target group of this course is primarily incoming graduate students and advanced undergraduates in BME or related disciplines including life science. The wide range of quantitative background of BME students is the main factor that sets this course apart from machine learning courses traditionally taught in other engineering and computer science programs. A significant proportion of incoming BME Masters students intend to use our BME program as a vehicle to enhance their preparation for future careers as data scientists in the biomedical industry. With this target group in mind, this course has no prerequisite and a paramount goal of this course is to provide students an appreciation of knowing the "why" and not just the "how" in biomedical data analytics. The course objectives (LOs) are summarized in Table 1. The course begins with the review of basic knowledge in probability, statistics, and programming in Python (Appendix A). The rest of the semester covers common supervised and unsupervised learning algorithms, and model evaluation (Fig. 2). This review article provides the description of each covered topic [1]. While these topics are similar to introductory machine learning courses offered by other Engineering and Computer Science departments and online courses [2]-[4], students learn to apply these algorithms to biomedical and life science applications. Students are exposed to different types of biomedical data such as measured physiological signals and medical images (Appendix B). They also see typical issues with biomedical data such as imbalanced datasets in rare diseases or datasets with many missing values. Feature extraction (certain characteristics of the signal or image) techniques are not discussed at length but are addressed per relevant example.

Assessment in this course includes homework (formative assessment: conceptual and coding problems), a midterm exam (summative assessment: conceptual and calculation problems) and a group project (summative assessment). The group project provides students with an opportunity to get hands-on experience of how one would approach a real-world biomedical problem of choice. The project involves literature review, securing and preparing dataset, implementing at least two ML algorithms and model evaluation (see Appendix C for more details). To assess students' performance, questions on all assessments are mapped to the corresponding LOs and topics. The scores earned from all problems mapped to the respective LO is converted to percentage of the total score of questions mapped to that LO. The percentage is then converted to 5 levels of mastery: 1 (below 60%): no understanding, 2 (60–70%): novice, 3 (70–80%): meeting expectation, 4 (80–90%): above expectation, and 5 (90% and above): accomplished. A similar calculation is applied to evaluate students' understanding per topic. Additionally, the number of questions mapped to each LO or topic is recorded to give a sense of assessment frequency and whether the frequency of assessment is related to performance.

Results: In Fall 2020 (hybrid class due to the COVID-19 pandemic), a total of 23 students enrolled in this course: 1 undergraduate senior, 18 Masters students and 4 PhD students. All but three (Chemical Engineering) students were BME students. Eleven students had undergraduate training in BME, 6 in other engineering majors or Physics, and 6 in Life Science (e.g., Biochemistry). The self-reported incoming programming skill levels were: 3 beginners (no programming background or can follow basic tutorials but need assistance in understanding the code logic), 19 intermediate level (understand the tutorial code logic and can adapt the examples to their own problems) and 1 advanced programmer (can implement algorithms from scratch while ensuring code optimization). The 5-point scale ratings of students' fulfillment of the LOs are shown in Fig.1. Students were rated at above expectation and accomplished levels on LO2, 3 and 6. One student (4%) was at the novice level for the ability to evaluate and justify the selection of learning algorithms and model selection (LO5). LO1 was assessed the most often throughout the semester, followed by LO4. The frequency of LO2, 3 and 5 assessment was relatively uniform (see Appendix D for the assessment frequency). The last objective (written skill) was assessed only four times throughout the semester through the project report. There was no correlation between the assessment frequency and students' performance by learning objectives.

Table 1: Course learning objectives

1.	Gaining the understanding of the basic concepts behind ML algorithms.
2.	The ability to identify data type, and prepare data for ML model.
3.	The ability to select suitable machine learning algorithms.
4.	The ability to implement ML techniques in Python.
5.	The ability to evaluate and justify the selection of learning algorithms and model selection.
6.	The ability to demonstrate effective scientific communication in written form.

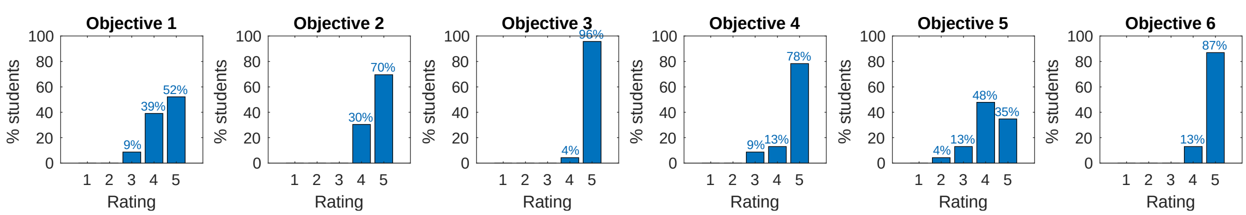

Fig.1 Percentage of students (of N = 23) fulfilling each course learning objective in 5-point scale

Figure 2 shows the 5-point scale ratings of students' understanding per topic. Students met the expected level of understanding in most topics. A few students did not develop good level of understanding in Naïve Bayes (9%), Model Evaluation (9%) and Perceptron (4%). Students showed relatively lower performance in Regression and SVM, likely due to insufficient mathematical background (Appendix E). The frequency of assessment on all topics was relatively uniform (approximately 10 times per topic), except for classification and regression trees (CART), cross-validation and neural networks that were assessed twice as often (Appendix D). CART was assessed often because it covers three subtopics: classification trees, regression trees and random forests. Cross-validation and neural networks are employed in most AI/ML applications and, thus, require more thorough discussion and assessment. Surprisingly, students showed the highest performance in the Neural Networks, which is considered an advanced topic. This indicates some correlation between the frequency of assessment and students' performance.

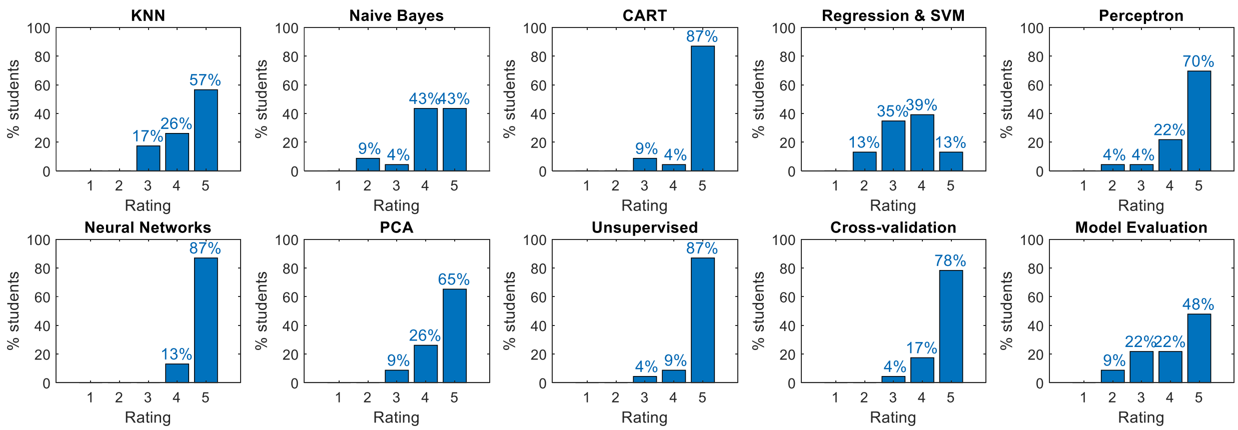

Fig.2 Five-point scale rating of students' understanding by topic. KNN: k-nearest neighbors, CART: classification and regression trees, SVM: support vector machine, PCA: principal component analysis.

Discussion: Overall, this iteration of the course successfully accomplished all course objectives at the level of mastery of meeting expectation and above. Students showed slightly lower LO5 performance (the ability to evaluate and justify the selection of learning algorithms and model selection), as it requires higher level of thinking according to Bloom's taxonomy [5]. Of particular interest to the author was LO4: the ability to implement machine learning in Python, which the author anticipated a large variation in performance by students due to their different levels in programming skills. However, most students showed competence in correctly implementing ML algorithms in Python language. This outcome may be explained by the optional resources provided for coding exercises, ranging from skeleton codes to short snippets to short hints, allowing students to choose the level of guidance they need. This suggests that the optional resources was appropriate for this target group, but a student perception survey should be conducted in the future course iteration to verify this observation. Lower performance per topic was generally due to insufficient background (Appendix E), which was mitigated during the semester by providing additional background materials and exercises. But this measure alone may not be adequate. For example, Regression and SVM, the first topics that require more advanced mathematical background, e.g., linear algebra and optimization, were taught at a normal pace and this turned out to be the most challenging topics for students. On a contrary, students performed well in neural networks, a more difficult topic with more allocated time and exercises. This underscores the importance of appropriate balance between the pace and the topic coverage. To ensure that students have solid understanding of the topics covered, more time should be allocated to more challenging topics. This means that the total number of topics covered may need to be reduced in the future e.g., unsupervised learning, because it is used not nearly as much as supervised learning in healthcare and biomedical applications [6]. In summary, given the diverse quantitative background of BME graduate students, the key to developing an introductory course in machine learning for BME students would be to emphasize the conceptual aspects of AI/ML and specialized issues connected with application to biomedical data.

References

[1] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *Sn Computer Science*, vol. 2, no. 3, p. 160, May 2021.

[2] S. Chandrasekaran, "BME 499.060: AI in BME", *Systems Biology Lab, Biomedical Engineering, University of Michigan*. Available: https://systemsbiologylab.org/ai-bme-syllabus.

[3] M. Gormley, "Introduction to Machine Learning", *Carnegie Mellon University*. Available: http://www.cs.cmu.edu/~mgormley/courses/10601/ schedule.html.

[4] A. Ng, "Machine Learning" offered by Standford, *Coursera*. Available: https://www.coursera.org/learn/machine-learning.

[5] L. W. Anderson and D. R. Krathwohl, *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Pearson, 2001.

[6] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Sheng and Y. Wang, "Artificial Intelligence in Healthcare: Past, Present and Future," Stroke Vasc Neurol, vol. 2, no. 4, p 230-243, Dec 2017.

Appendix A: Probability, statistics and Python review

Given that students come from various undergraduate training, the first week of class is used to review some basic knowledge necessary for the course. Probability and statistics review topic covers probability calculation and properties, discrete vs. continuous variables and probability mass function vs. probability density function, joint, marginal and conditional probability, independence and conditionally independence, expected value, variance and standard deviation and briefly on types of probability distributions: Bernoulli, binomial, multinomial and Gaussian distribution. Introduction to Python includes creating different data types, importing libraries, importing, summarizing, visualizing dataset and basic mathematical operations.

Appendix B: Datasets

Datasets used in this course include data from publicly available sources, e.g., UCI Machine Learning Repository, Kaggle, Data.gov, the instructor's own research data and synthetic data. These datasets may contain raw time-series signals (e.g., ECG and EEG signals), images (e.g., blood cells) and/or the already derived features and calculated index from the raw data (e.g., features derived from cancer cell autopsy).

Appendix C: Group project

Students may choose any biomedical-related problems to apply ML to help answer the research questions. Students are required to implement at least two appropriately-selected ML algorithms to the problem, then evaluate the ML models. The first step after identifying the problem of interest is to review current ML literatures on a similar problem where students familiarize themselves the kind of data used in such application and identify the ML algorithms used. Next, students secure the dataset (from publicly available sources or their own research data) and compare their dataset to the ones used in the literatures. This step should give them an idea of data preparation needed to be performed on their own dataset and how complete/incomplete their dataset is. Students then choose at least two appropriate ML algorithms, which may be the same or different to ones reported in the literatures. Lastly, students evaluate the models by comparing the two (or more) implemented models as well as comparing their results with the literatures. Students must also identify the strengths and weaknesses of their selected methods.

Appendix D: Frequency of assessment per learning objectives and topics

Each question in all assessments (homework, midterm exam and project) are mapped to the corresponding learning objective (LO) or ML topic. The number of times the questions mapped to the LO/topic are asked in one semester is taken as the frequency of assessment.

By learning objectives:

LO	Description	Frequency
1	Gaining the understanding of the basic concepts behind ML algorithms.	60
2	The ability to identify data type, and prepare data for ML model.	19
3	The ability to select suitable machine learning algorithms.	18
4	The ability to implement ML techniques in Python.	27
5	The ability to evaluate and justify the selection of algorithms and model selection.	16
6	The ability to demonstrate effective scientific communication in written form.	4

By topics:

Algorithm	Frequency
k-nearest neighbors (KNN)	12
Naïve Bayes	8
Classification and regression trees (CART)	22
Regression and SVM	11
Perceptron	8
Neural networks	19
Principal component analysis (PCA)	11
Unsupervised learning (k-means and hierarchical clustering)	8
Cross-validation	18
Model evaluation	8

Appendix E: Students' background

This course has no prerequisite as it is designed to serve as a gateway for students to get into data scientist career path in the biomedical industry, regardless of their quantitative background. Based on students' performance reported in this work, the following topics are recommended to be covered more extensively either as refresher, ramp-up or in-class materials: joint and conditional probability, independence and conditionally independence, conceptual understanding of the cost function, conceptual understanding of derivatives, the concept of optimization with derivatives, vector and matrix notations, vector and matrix operations, linear algebra (e.g., unit vector and vector projection for SVM). While these topics are "math-heavy", the goal of this course is not to teach students to show proof or derive complex mathematical functions. Rather, the focus is on teaching students to understand conceptually what these mathematical expressions and operations mean and what are their significance.