USING ARTIFICIAL INTELLIGENCE TO PROVIDE DIFFERENTIATED FEEDBACK AND INSTRUCTION IN INTRODUCTORY PHYSICS

by

Jeremy Munsell

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Physics and Astronomy

West Lafayette, Indiana

May 2022

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Sanjay Rebello, Chair
Department of Physics and Astronomy
Department of Curriculum and Instruction

Dr. Carina Rebello
Department of Physics and Astronomy

Dr. Andrew Hirsch
Department of Physics and Astronomy

Dr. Dan Milisavljevic
Department of Physics and Astronomy

Approved by:
Dr. Gabor A. Csathy

ACKNOWLEDGMENTS

I would like to dedicate this dissertation to my family, especially my mom and dad, for working hard to provide me the opportunity to pursue my studies. I can't say enough to express my gratitude for all of the sacrifices you've made over the years. It has been a long road, but we've finally made it. I would also like to thank my brother Jeff and my sister Anj. I wish to honor the memory of my grandparents. I know grandma Griggs would be so proud and I'm sorry she was not here to see this. Also, I would like to acknowledge my grandma Norma, and my grandpa Ken. Grandpa Ken raised a family that was always taught to value education. My desire to pursue this level of education is directly attributable to him.

I would also like to give my sincerest thanks to my advisor Sanjay. I never could have guessed that people as generous and supportive as him exist in academia. You have given me the tools to succeed in my studies, and the patience and support to help me see it through. I truly wish to thank you from the bottom of my heart. I also acknowledge help and mentoring from Dr. Tianlong Zu.

I also wish to thank my wife, Trang. This has been a very difficult time in our lives. I know you made this time easier for me, and I hope I have made it a little easier for you. I look forward to what comes next, with you by my side. I also wish to thank my second parents, ba mẹ. Con cám ơn ba mẹ đã lo lắng, quan tâm, và chăm sóc tụi con.

To Sandy and Janice. You are both amazing! Over the years you have helped so many students find our way when we are lost. You both are like mothers to the department. Thank you so much!

Finally, I wish to thank my friends Kevin, Andres, Cheng-An, Terrence, Alireza, Honggu, Anna, and Susan. You all have made grad school a very enjoyable experience. Adeel, our hearts were broken when we lost you. You will forever be in our memories.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

μ	average value
σ	standard deviation
∇	gradient operator
$\boldsymbol{w}$	weight vector
w_0	bias
$\boldsymbol{\theta}$	combined weight/bias vector
$J(\boldsymbol{\theta})$	loss (error) function
α	learning rate
A^T	transpose of matrix
$\sum$	summation of a series
$\prod$	multiplication of a series
$\boldsymbol{\pi}$	irrational number pi
e	base of the natural logarithm
$\langle a \rangle$	the average of a
g	normalized gain
$P(A)$	probability of event A
$P(A\|B)$	probability of event A given event B
α^*	significance threshold

ABBREVIATIONS

WMC	working memory capacity
CLT	cognitive load theory
ICL	intrinsic cognitive load
ECL	extraneous cognitive load
GCL	germane cognitive load
ERE	expertise reversal effect
HLG	high level guidance
LLG	low level guidance
HPK	high prior knowledge
LPK	low prior knowledge
HS	high school
AP	advanced placement
GPA	grade point average
ML	machine learning

ABSTRACT

Cognitive load theory (CLT) lays out a tripartite scheme concerned with how learners cognitively interact with instructional materials during learning and problem solving. Cognitive load refers to the utilization of working memory resources, and CLT designates three types of cognitive load as intrinsic cognitive load, extraneous cognitive load, and germane cognitive load. Intrinsic cognitive load is related to the intrinsic complexity of the material. Extraneous cognitive load is concerned with unnecessary utilization of cognitive resources due to suboptimal instructional design. Germane cognitive load results from processing the intrinsic load and schema acquisition. The expertise reversal effect follows as a consequence of CLT.

The expertise reversal effect (ERE) states that instructional materials that are beneficial to low prior knowledge (LPK) learners may be detrimental to high prior knowledge (HPK) learners. Less guided materials have been shown to reduce extraneous cognitive load for these learners and therefore produce a greater benefit.

In this work we present the development of online instructional modules that deliver content in two distinct styles, differentiated by their use of guiding features. the high level guidance version (HLG) uses guiding features, such as animations and voice narration, which have been shown to benefit LPK learners. Alternatively, guiding features have been shown to be destructive to the learning of HPK students. The low level guidance (LLG) version uses text in place of voice narration and pop-up content in place of continuous animations. Both versions led to a statistically significant improvement from pre-test to post-test. However, both HPK and LPK students showed a preference for the HLG version of the module, contrary to the ERE. Future work will focus on improving the ability to indentify HPK and LPK students, and refining methods for providing optimal instructional materials for these cohorts.

Meanwhile, the use of machine learning is an emerging trend in education. Machine learning has been used in roles such as automatic scoring of essays in scientific argumentation tasks and providing feedback to students in real time. In this work we report our results on two projects using machine learning in education. In one project we used machine

learning to predict students' correctness on a physics problem given an essay outlining their approach to solving the problem. Our overall accuracy in predicting problem correctness given a student's strategy essay was 80%. We were able to detect students whose approach would lead to an incorrect solution at a rate of 87%. However, deploying this model to provide real-time feedback would necessitate performance improvement. Planned future work on this problem includes hand grading essays to produce a label that reflects the scientific merit of each essay, using more sophisticated models (like Google's B.E.R.T.), and generalizing to a larger set of problems.

In another study, we used data about students' prior academic behavior to predict academic risk in a first-year algebra based physics course. Their final course grade was used to define their risk category as; B- and above is designated low risk, and C+ and below is designated as high-risk. Using a mix of numerical and category features such as high school gpa, ACT/SAT scores, gender, and ethnicity we were able to predict student academic risk with 75% overall accuracy. Students with a very high grade (A) or students with a very low grade (D,F,W) were identified at a rate 92% and 88% (respectively).

Prior work [1], [2] has shown that performance can be greatly increased by including in-class features into the model. Future work will focus on obtaining raw data, rather than using curved scores reported to the university registrar. Also, obtaining more batches of data to improve predictive power with existing models developed in this study.

1. INTRODUCTION

Cognitive load theory (CLT) provides insight into the cognitive architecture available to students and how it is used during learning and problem solving [3]–[5]. CLT frames a triarchic model where cognitive load is delineated as intrinsic cognitive load, extraneous cognitive load, and germane cognitive load. Intrinsic load originates from the essential complexity of the material to be learned. Extraneous load stems from suboptimal instructional design and forcing the learner to split their attention between multiple sources of information. Germane load is related to processing the intrinsic load and schema formation. CLT provides guiding principles for the design of instructional materials. If the cognitive load imposed by instruction is too high, working memory resources will be overwhelmed and learning will not take place.

Furthermore, it has been demonstrated that students struggle to develop problem solving skills [6], [7]. The traditional wisdom of ‘just solve a bunch of questions, then you’ll be an expert problem solver’ doesn’t work like we think it does. We now know that appropriately designed worked examples are superior in facilitating the acquisition of domain knowledge relative to conventional problems which do very little to help novices become experts [5], [8]. Properly designed examples include guiding features that demonstrate expert-like thought in a step-by-step fashion to reduce extraneous load by focusing the learner’s attention on problem solving moves that are productive towards a solution [5].

It should be noted that worked examples of this nature do not benefit all learners equally. Work on the expertise reversal effect shows that highly guided instructional approaches, while being beneficial for novice learners, do not have the same benefit to high knowledge learners [9]–[11]. In fact, high prior knowledge learners perform better with a lower level of guidance because the guiding elements that reduce intrinsic load for novices increase extraneous load for high knowledge learners [12]. In order to present instructional materials in a way that is optimal for all students, their individual learner characteristics must be respected.

Current e-learning systems are very useful for certain things like managing a course with a large number of students, but they are not effective at fostering problem solving ability [13]. This is truly a wasted opportunity as online instruction systems provide an interactive environment that allows students to move through difficult material at their own pace. For an e-learning system to help everyone, it should be adaptive for students with different knowledge levels. In this document, I present my progress towards the development of an online learning system. I also present supplemental work in using machine learning to classify students based on their individual characteristics. Specifically I used prior academic and demographic variables to classify students into one of two groups. The two groups were defined based on the final letter grade in a first-year algebra based mechanics course. However, the groups could serve as a proxy for domain knowledge level. In another supporting project, students solved the classic ballistic pendulum problem on an online quiz during lab. As part of their solution, they were asked to write a short essay outlining the approach they used to solve the problem. This essay included the assumptions they made, the principles of physics they used, and how they were used.

Machine learning refers to a set of algorithms that are capable of carrying out tasks as a human would without explicitly being programmed to do so. Instead of explicit instructions, machines learn as they are exposed to labelled training data (in the case of supervised classification), or by recognizing statistical trends in un-labelled data (in the case of unsupervised learning).

Machines can be leveraged to perform tasks including but not limited to classification, regression, and generation of speech. These tasks can be performed on data such as numerical tabular data, text data, and image data (again including but not limited to).

In my work, I endeavored to combine the power of artificial intelligence with novel techniques for physics education.

Our main research questions were the following:

1. To what extent will students benefit from interactive online modules that are made adaptive to student's domain knowledge level?

2. To what extent can we classify students based on their level of domain knowledge using only data that is available prior to the start of a course?

3. Given a student essay outlining their strategy for solving a physics problem, with what degree of accuracy can I predict whether the student will get the problem correct?

In study 1, we addressed the first research question by developing an online instructional module on the subject of force and motion. Specifically, the concepts of vector analysis, drawing a force diagram given a problem statement, writing Newton's second law in terms of vector components, and solving Newton's second law. This module was a web application created from scratch using the django framework in Python, and hosted on a web server. Students accessed the module remotely through the web browser on their computer. The module was intentionally designed not to be usable with mobile devices to reduce distractions from other applications (e.g. facebook notifications), and ensure that the screen was large enough to properly display the content. The webapp was utilized by students from a first year, algebra based mechanics course. It was available in two versions.

The module was made of three sections. There was a pre-test, an instructional portion, and a post-test. The pre/post test were isomorphic in the sense that conceptual questions were held fixed, and numerical questions had changed numbers between pre/post while all other aspects were unchanged.

The two versions were differentiated by the instruction phase of the module. In one version, designated as high level guidance (HLG), material was delivered in the manner of a lesson being taught by an instructor. Concepts were illustrated through pictures and animated scenes. Animations were used to demonstrate concepts such as vector addition. These scenes and animations were synched to voice narrations produced by an AI generated voice. The overall presentation was similiar to a lecture video. However, all the elements were made to render in the browser to provide a higher level of immersion for the student. The trajectory of the lesson was more-or-less fixed, however they were allowed to go back in the event they missed something.

The low level guidance (LLG) module differed from the HLG variant by presentation only. The LLG version delivered content statically, without using animations or narration. Users are presented with prominent content areas labelled by concepts that are relevant to the lesson.

When the user hovers their mouse over these content areas, pictures or diagrams appear on the screen along with text explaining the figure. When the user moves their cursor out of the content area, the text and figure disappears from the screen. This method of displaying content is an interactive alternative to a static slide or image. Furthermore, even though the order of the content areas suggest a path through the lesson by the learner, this mode allows the user more independence as they move through the lesson. The learning modules will be described in more detail in chapter 4.

Research question 2 was explored in study 2 by obtaining data from the University registrar for students in a first year, algebra-based mechanics course. This sample of students were mainly from the school of technology, pre-pharmacy, and the life sciences. This data included current and prior academic measures, and demographic attributes. Among the academic data was students' high school (HS) grade point average (GPA), most-recent-prior-term college GPA, ACT/SAT scores, and advanced placement (AP) math and physics scores. The demographic information included gender, ethnicity, and first generation college student status.

Also included in this data was the students' final course grade. The final course grade was used to divide the students into two classes. One class (class 1) consisted of students whose final grade was 'B-' or above, and the other class (class 0) was students whose final grade was 'C+' or below. A machine learning model was trained using this combination of data and labels from several consecutive semesters. The trained model was used to predict the final course grade for students from the most recent semester for which the data was available.

I investigated research question 3 in study 3 by embedding a question in an online quiz taken during lab in a first year course for future scientists and engineers over two non-consecutive semesters. The quiz question asked students to solve a variation of the classic "ballistic pendulum" problem. As part of the question, they were asked to write an essay outlining their approach to solving the problem. The essay was to include a discussion of the principles of physics used in the solution, the assumptions that were made, and how these were used.

The correctness of the answer to the problem was used as a binary class label, with '1' denoting a correct final answer and '0' indicating the final answer was incorrect. The essays were transformed into a high dimensional vector using the methods of natural language processing. This combination of essays and labels from one semester were used to train a machine learning model, and the trained model was used to predict the problem correctness of the other semester given student essays.

1.1 Overview

This document will be organized in the following fashion. In the second chapter we will discuss cognitive load theory in more detail as well as a providing a short list of some of the cognitive load effects that follow from cognitive load theory proper. We will go on to discuss some of the features of adaptive learning systems and some principles of design. One of the central tasks of a successful adaptive learning platform is being able to appropriately classify students based on their knowledge level. The use of machine learning could facilitate this task. Consequently, a discussion of what machine learning is, how it is used, and how some of the algorithms work will follow.

In the third chapter we will discuss some preliminary work that was performed to address the research questions outlined in this chapter. In the fourth chapter, I lay out the development of and deployment of an instructional module. I will provide a deeper discussion of the differences between the two versions of the module, and give some results about how students benefitted from the use of this module in the context of research question 1. In chapter 5 we describe the project the use of machine learning to predict student outcomes in a first year course in mechanincs. Specifically, we wil discuss the data that we obtained, how the data was prepared, how our machine learning model was built and validated, and provide some discussion about the implications of this work for improving student outcomes and the ability to classify students based on their level of domain knowledge. In chapter six I will outline the work that was conducted to use natural language processing to assess student scientific argumentation. Specifically, I disucuss the data that we collected, how it was used in the study, the results that I obtained, and the implications for future studies.

2. BACKGROUND

In this chapter I describe the motivating principles behind my work. I will start with an introduction to cognitive load theory which provides insight into the cognitive architecture available to students and how they use it during learning and problem solving. This will be followed by discussion of some of the effects that follow as a direct result of cognitive load theory, for example the expertise reversal effect. Then, we will discuss adaptive learning platforms where instructional content is dynamically delivered in a way that is optimal for individuals. We will end this chapter with a discussion of machine learning and natural language processing.

2.1 Cognitive Load Theory

In this section we introduce a framework, cognitive load theory, which relates cognitive architecture to instruction and learning. Cognitive load theory provides guiding principles which serve as a basis for the design of instructional materials that are optimal with respect to the cognitive resources available to the student.

Human memory and the interplay between long-term memory and working memory is central to cognitive activities such as learning and problem solving. The multicomponent model of working memory proposed by Braddeley and Hitch [14], [15] specifies working memory as consisting of four distinct parts: the central executive, the episodic buffer, the phonological loop, and the visuo-spatial sketchpad.

The central executive (Fig 2.1 (1)) is the control center of working memory. It assigns tasks the other systems involved in learning and memory tasks, and coordinates information when multiple tasks are performed simultaneously. The central exectutive also performs high-order cognitive tasks such as problem-solving. Moreover, the central executive is responsible for directing attention to relevant information and discarding the irrelevant.

The phonological loop (Fig 2.1 (2)) processes linguistic information and includes a storage mechanism that allows one to keep information active in memory by rehersal (repeating to yourself). The visuo-spatial sketchpad (Fig 2.1 (4)) processes and stores visual information

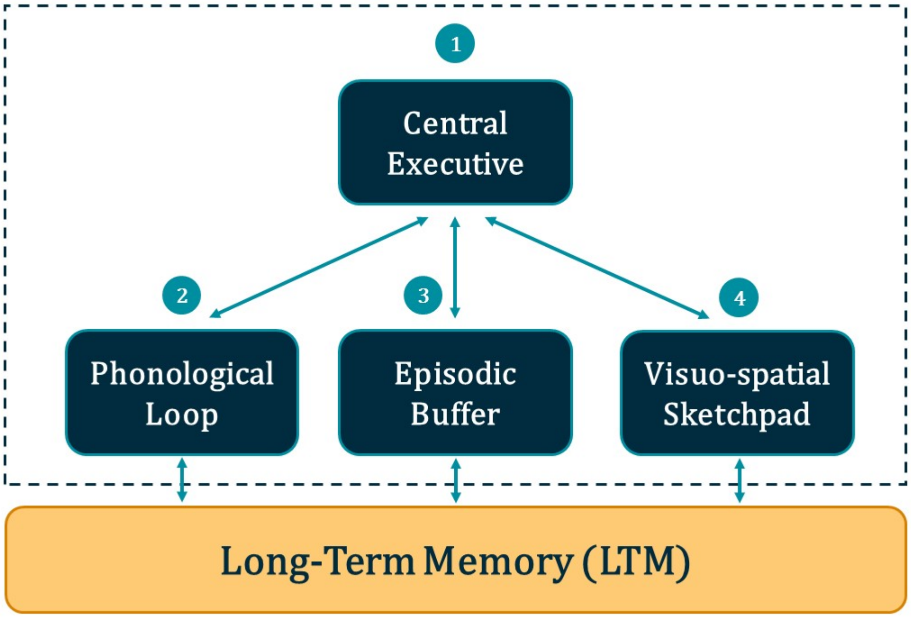


Figure 2.1. Working memory is described as consisting of 4 parts. [14], [15] The central executive (1), the phonological loop (2), the episodic buffer (3), and the visuo-spatial sketchpad (4)

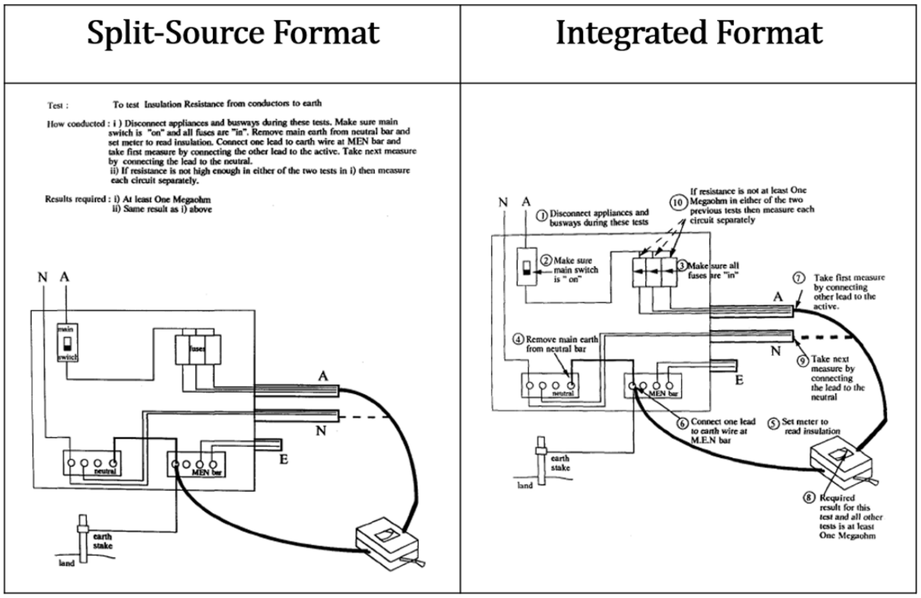
(such as the visual attributes of an object) and spatial information (such as mental maps). The episodic buffer (Fig 2.1 (3)) serves to synthesize information that has both a linguistic and visuo-spatial component, and acts as a temporary cache. [15].

Cognitive load theory (CLT) describes how working memory resources are utilized to process information in learning and problem solving. CLT frames a triarchic model, enumerating cognitive load as intrinsic cognitive load, extraneous cognitive load, and germane cognitive load. Intrinsic cognitive load, hereafter referred to as intrinsic load, is related to the learner and the material that must be learned. Extraneous cognitive load, hereafter referred to as extraneous load is related to the presentation of the material and is therefore sensitive to instructional design. Germane cognitive load, hereafter referred to as germane load is the cognitive load associated with processing the intrinsic load and constructing or extending schema [5].

CLT was introduced in 1988 and has been continually revisited and refined in terms of its explanatory power and resolution since. Early versions of CLT place high importance of schema acquisition as the primary distinguishing factor between novice and expert-like problem solvers. A schema is defined as a cognitive structure enabling problem solvers to recognize problems as belonging to a category of problems which normally require a common set of steps to solve. Schemas reflect domain-specific knowledge, which novices do not possess [3].

Expert problem solvers tend to group problems based on their solution strategy (e.g. conservation of energy), where novices generally group problems based on surface features (e.g. block problems, car problems, etc…) [16]. In this earliest version of CLT schema acquisition is the primary goal of learning as other, less-directed approaches to problem solving, impose a higher cognitive load by spending mental resources on activities that are not productive to learning. Here, Sweller alludes to the notion of extraneous cognitive load without making it concrete [3].

In a later version of CLT, the model is refined to explicitly refer to the cognitive load imposed by suboptimal instructional design that is detrimental to learning as extraneous load [17]. For example, providing information in a split-source format (such as using words and a diagram) where neither is intelligible on its own requires the learner to split their attention and mentally integrate the mutually referring information in the text and the diagram. Since this activity of mental integration is not necessary for learning and uses up resources on activities that not productive to learning it constitutes an extraneous load. A more friendly format from an extraneous load standpoint is an integrated format where relevant text is integrated into the diagram and the learner is no longer required to split their attention between different sources of information (Fig 2.2). The authors consider a middle ground in a series of experiments where they seek to determine if redundant information which doesn't require mental integration is destructive to learning. A study was conducted with 28 first year electrical engineering apprentices. The apprentices were trained in testing newly installed electrical equipment with either a conventional split source diagram (Fig 2.2 frame 1) or a modified integrated format diagram (Fig 2.2 frame 2). After training, the apprentices were given a practical test. The results of those experiments provide strong evidence that presenting information in a modified format that shows the material in an integrated manner is superior to a split-source presentation where complete information is



Test : To test Insulation Resistance from conductors to earth

How conducted : i) Disconnect appliances and busways during these tests. Make sure main switch is "on" and all fuses are "in". Remove main earth from neutral bar and set meter to read insulation. Connect one lead to earth wire at MEN bar and take first measure by connecting the other lead to the active. Take next measure by connecting the lead to the neutral.
ii) If resistance is not high enough in either of the two tests in i) then measure each circuit separately.

Results required : i) At least One Megaohm
ii) Same result as i) above

Split-Source Format
Integrated Format
N
A
main
switch
fuses
A
N
E
neutral
MEN bar
earth
stake
land
N
A
① Disconnect appliances and busways during these tests
② Make sure main switch is " on"
③ Make sure all fuses are "in"
④ Remove main earth from neutral bar
⑤ Set meter to read insulation
⑥ Connect one lead to earth wire at M.E.N bar
⑦ Take first measure by connecting other lead to the active.
⑧ Required result for this test and all other tests is at least One Megaohm
⑨ Take next measure by connecting the lead to the neutral
⑩ If resistance is not at least One Megaohm in either of the two previous tests then measure each circuit separately
A
N
E
neutral
MEN bar
earth
stake
land

Figure 2.2. The figure shows information presented in a slit-source format (frame 1) and an integrated format (frame 2) [17].

presented in multiple sources. This result confirms that splitting attention and (unnecessarily) mentally integrating separate sources of information produces an extraneous load and reduces learning [17].

Sweller [4] elaborates on schema acquisition as a primary goal of learning by enumerating two critical mechanisms. In one mechanism schema acquisition occurs gradually as the learner is presented with new information. Subject knowledge is organized into schemas as new information is altered to make it congruent with the learner's existing knowledge. Another mechanism relevant to information processing during learning and problem solving is automation of schema. Information is either processed in an automatic or a controlled manner. Controlled processing occurs when a learner must consciously attend to information. Automatic processing occurs when a learner does not need to deliberately focus attention on information during handling. Consider reading as an example which illustrates the distinction between automatic and controlled processing. Reading the words on this page is automatic, while trying to understand the meaning of a passage is controlled. The switch from controlled processing to automatic processing is continuous and slow as familiarity with a given domain is obtained [17]. Fig 2.3 shows that when a learner with relevant domain knowledge is presented a problem, they must keep the problem in their working memory while searching long-term memory for relevant schema. If the schema is not automated, then they activate several possible candidate schemas and must use working memory resources to select the correct one. Alternatively, if the schema is automated, the correct one is automatically activated. If the learner lacks domain knowledge then all of the processing takes place in working memory and can easily overload the working memory capacity.

Schema acquisition and automation both serve to reduce cognitive load. The number of elements that can be accommodated in working memory is fixed at around 7 [15]. An element can be defined as a unit of information. Schemas reduce cognitive load by aggregating elements into fewer units containing more information per unit. Schema automation reduces cognitive load by significantly reducing the amount of processing needed in working memory to activate relevant schemas.

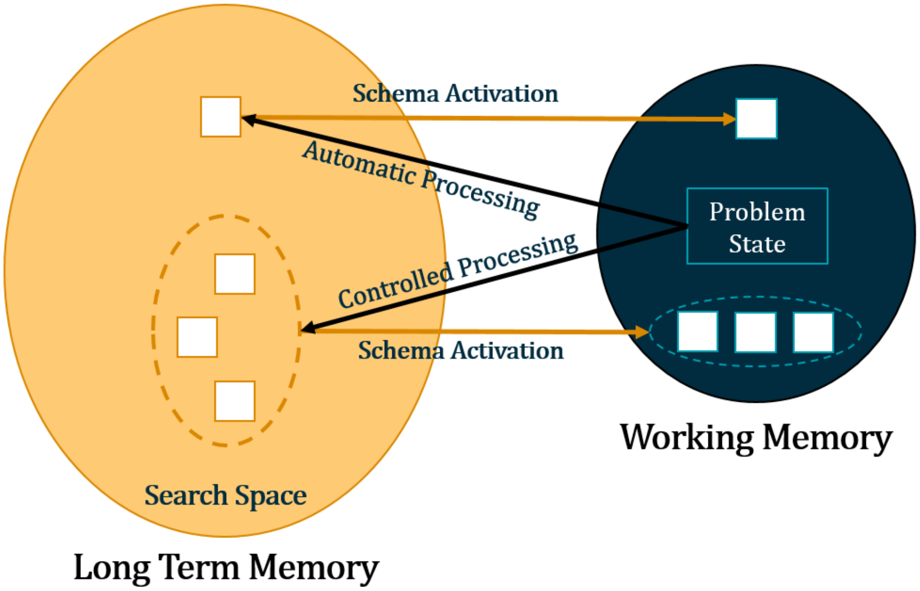


Figure 2.3. Controlled processing results in a search of LTM for relevent schema and WM resources must be used to determine the applicable schema. Automatic processing bypasses the search and relevant schema are automatically activated.

An element can be anything that needs to be learned or has been learned such as facts, concepts, formulas, definition, or even schemas themselves. What constitutes an element depends on the learner and the material to be learned. A learner with high domain knowledge in a content area will tend to group information in larger chunks so that each element contains more information than the elements of their low domain knowledge counterparts.

Interaction of elements occurs when elements must be processed simultaneously rather than sequentially. To appreciate what is meant by interacting elements, consider the following two tasks. Memorization of events and dates in a history class and solving an algebraic equation. For the first, pairs of dates and events can be memorized as independent pair elements with only slight interaction associated with the order of date/pair elements (e.g. the civil war happened before the Vietnam war). For the second, the learner must consider the legal operations, the order in which we apply them, and the scope of each operator.

Interaction of elements is the primary determinant of intrinsic cognitive load [5]. If the number of interacting elements in a content are low, then there will be a low intrinsic load associated with it. It would follow that memorizing dates/events in a history class would impart a low intrinsic load whereas solving an equation would impart a higher intrinsic load. That is of course unless the problem solver possesses a schema for solving that type of equation. In that case, the schema for solving that equation would itself be an element and the interaction would be internal to that element (analogous to internal forces in a momentum conservation problem). This in turn shows how schema acquisition reduces cognitive load by reducing the number of interacting elements.

CLT was later refined in a stroke of elegance to provide a firm unifying theory for the different types of cognitive load, where element interactivity was designated as the fundamental currency. In this version of CLT element interaction is still assumed to be the principal determinant of intrinsic load. Estimating the number of interacting elements must simultaneously account for the information to be learned and the knowledge level of the learners [5].

Working memory load is not only generated by the complexity of the material, it can also originate from less than optimal design of educational materials. Extraneous load is also induced through interacting elements by way of integrating redundant information. A litmus test for distinguishing between intrinsic and extraneous load, since they have common origins, is that if element interactivity can be reduced without altering the nature of what is learned, then the cognitive load is extraneous. If the number of interacting elements can only be modulated by changing what is learned, then the load is intrinsic [5].

Germane load also originates from interacting elements, but it has a different status than intrinsic or extraneous load. Intrinsic load and extraneous load are both related to the material to be learned, whether it is the natural complexity or the presentation of the material. Germane load, however, is entirely due to the characteristics of the learner.

The combination of intrinsic load and extraneous load constitute the total cognitive load. Germane load, on the other hand, does not represent an independent source of cognitive load, as it is related to both intrinsic and extraneous load. It refers to the cognitive resources that are available to the learner to deal with the intrinsic load associated with the material. Extraneous load depletes these available resources. If intrinsic load is high and extraneous load is low, then germane load will be high as the learner is able to devote sufficient resources to processing the intrinsic load and learning will take place. Meanwhile, for the same level of intrinsic load, if extraneous load is high, then the germane load will be lowered and learning will be reduced.

This formulation suggests the following mathematical relation:

$$GCL = WMC - ECL \tag{2.1}$$

Learning can only occur if the GCL 2.1 is greater than or equal to the ICL associated with that material. This formulation assumes that the learner will devote all resources to leaning, irrespective of the task, a more reasonable relation was proposed and validated by experiment [18].

$$GCL = min(ICL, WMQ - ECL) \tag{2.2}$$

Assuming a constant level of motivation, i.e. the learner devotes available resources to learning, the learner has no direct control over germane cognitive load. Rather, it is manipulated indirectly by changes in intrinsic load and extraneous load.

2.2 Element Interactivity Effect

Many educational effects emerge as implications of cognitive load theory. These various effects are mechanistically specified in terms of element interactivity. Infact, the element interactivity effect states that in order to observe any of these effects, a necessary condition is that the intrinsic load must be sufficiently high. [19]. The element interactivity effect is treated as a separate effect because the cognitive load effects stemming from intrinsic load have been found to have deep implications for cognitive load effects related to extraneous load, and both are based on interacting elements.

2.3 Goal Free Effect

Problem solvers learn more from solving problems with a reduced or eliminated focus on a singular goal relative to a conventional problem [20]. Conventional problems can be solved by a novice using a mean-ends strategy. A means-ends strategy approach to a conventional problem requires carrying information about the problem state, the goal state, problem solving operators to reduce the differences between the two, as well as any intermediate goals. Goal free problems, on the other hand, only require encoding a problem state and legal operators. These problems also aid in schema acquisition which reduce cognitive load via a reduction in interacting elements [5].

2.4 Worked Example Effect

The worked example effect shows that worked examples are superior to their equivalent conventional problems [21]. To solve a conventional problem, while not possessing relevant schema, requires handling many interacting elements associated with a means-ends strategy. Meanwhile, a worked-example takes the learner from one step to the next by applying appropriate operations in the manner of an expert, and thusly reduces extraneous cognitive load by not wasting working memory resources dealing with interacting elements associated with legal but irrelevant moves [5].

2.5 Expertise Reversal Effect

The expertise reversal effect states that when considering two instructional procedures, the procedure that is successful for novices becomes less successful as their expertise increases. With further expertise acquisition, the procedure that was less effective for novices will become more effective for experts, and the procedure that was more effective for notices will become less effective for experts.

When presented with a novel task, novices lack requisite domain knowledge. Consequently, they regard this new information as a discrete set of interacting elements, which can easily exceed their available working memory resources. Expert learners package information into fewer elements, so that the same information occupies fewer working memory resources.

The expertise reversal effect can be specified in terms of element interaction. For given information, higher expertise levels reduce the level of element interactivity as learners can combine multiple elements into a single element. Instructional procedures designed to reduce working memory load in a high element interactivity task for novices are no longer effective for expert learners who already find themselves in a low element interactivity environment [22].

The expertise reversal effect necessitates a categorical change in interacting elements. Initially, a given set of interacting elements constitutes an intrinsic load to a novice learner, since they are essential to understanding the material. As the learner gains expertise, interacting elements that arise from components of educational materials that are intended to provide guidance to low domain knowledge learners now amount to an extraneous load since they are no longer necessary for apprehension of the information [5].

2.6 Prior Work on The Expertise Reversal Effect

The expertise reversal effect emerged from a multitude of studies. The interaction between instruction and learner characteristics was observer as early as the 1950s [9], [23], [24]. Multiple works noted that the benefit of instructional formats for low prior knowledge learners were absent for high prior knowledge learners [10], [11], [25]. A full expertise reversal effect was not documented until 2000 [12]. It appears that most or all of the research on the expertise reversal effect uses a within-subjects design where the effectiveness of educational formats is compared for the same subjects before training (low knowledge) and after training (high knowledge).

In one experiment N = 60 trade apprentices with three months of relevant training were to be trained on the use of a piece of industrial equipment using different instructional formats [12]. The participants were randomly assigned to one of four conditions: a diagram with visual text, a diagram with narration, a diagram with both text and narration, or the diagram only. After studying the diagrams in their respective experimental conditions, they were given a performance test. The diagram with text and narration significantly outperformed the other conditions. After a period of training including worked examples on how to use the diagrams, the difference between these conditions was greatly reduced with a reversal in effectiveness between the diagram only and the diagram with narration conditions.

In another study with N=70 trade apprentices in a major Australian manufacturing company, with 1.5 months of technical training, the effect of worked examples was compared with guided exploration on training with diagrams [12]. Participants were randomly assigned to either a worked example condition or a guided exploration condition. In the worked example condition, participants were shown how to use the diagrams to answer example questions. In the exploratory condition, participants were (minimally) guided through use of the diagrams by a computer program that asks them questions involving the diagrams. The worked example condition initially outperformed the exploratory condition, but after additional training the effect was reversed.

In a study comparing the effects of direct guidance and guided exploration, N = 40 introductory algebra students from Carnegie Mellon University were randomly assigned to one of four conditions: a verbal direction condition where students received non-specific (general) instructions, a direct demonstration condition where students were given specific instructions, a condition that received both verbal directions and direct demonstration, and a condition that received neither (discovery condition) [26]. The students solved 174 problems spread amongst four chapters. This work quantifies the performance of the different conditions as the number of transformations per problem, the time per transformation, the number of operator errors, and the number of transformation errors. The discovery condition underperformed the other conditions on the first few problems of every set according to every metric. For the rest of the problems in each set, a reversal in the performance was observed with expertise.

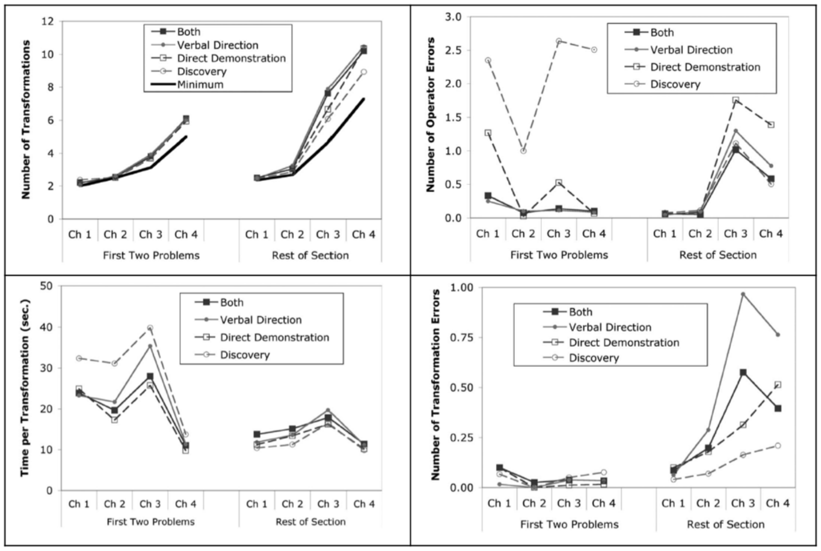


Figure 2.4. The plots shows the discovery condition underperforming the other conditions on the initial problems and outerforming the other conditions on the remaining problems in (almost) every set [26]

2.7 Adaptive Learning

In this section we discuss adaptive learning and the principles of designing adaptive learning systems. This fits into my thesis because I envision the modules described in the background section as being part of an adaptive learning platform where dynamic content is shown to students based on their knowledge level ala the expertise reversal effect.

Online learning (eLearning) systems are ubiquitous in today's educational landscape. While eLearning offers the advantages of ease of access to a worldwide audience, it poses several challenges to the learner. One of the primary challenges lies in the fact that currently used eLearning systems do not afford the learner the individualized attention that they may need to facilitate their learning. This deficiency of eLearning systems is particularly evident with regards to fostering problem solving ability [13]. Problem solving is an important skill that is valued in STEM education. Yet, students face several challenges with problem solving [7], [27].

Adaptive learning systems present material in a dynamic way that respects the characteristics of individual learners. Adaptive learning platforms follow three basic criteria. First, e-learning has the ability to operate in real-time; storing, retrieving, and using information. Second, content is delivered to users via their device using internet technology. Third, they transcend traditional educational methodologies by delivering content in a dynamic manner [28].

There are multiple approaches to adaptive learning such as the macro-adaptive approach where the rate of material delivery is adapted to suit individual needs [29], and the aptitude-treatment interaction approach where learners aptitudes are taken into account, but the learner also has direct control over how they interact with the content. This section will focus on the micro-adaptive approach. The micro-adaptive approach involves classifying learners according to their characteristics (abilities, motivation, knowledge, preferences) in order to provide them with the most appropriate educational experience [30]. This is approach requires analysis and monitoring of the learner's behavior and interactions with the system in order to adapt the pedagogical flow (such as the pace and style of delivery of content) of the learning experience [31].

This approach requires two primary procedures. First, the student must be characterized in terms of their abilities, motivation, knowledge, and etc. Second, this information must be used to optimize the delivery of content for that learner with those characteristics [30].

The necessary task of gathering and using data in real time to make decisions about how content should be displayed to students is facilitated by the use of machine learning. By gathering data from the student such as performance on assessment questions, timing data about how they interact with questions and videos, and even directly asking the students about their level of comfort with the material machine learning can be used to.

2.8 Machine Learning

We begin our discussion of machine learning with the observation that humans learn from their past experiences (especially mistakes) whereas machines traditionally require an explicit set of instructions provided by humans. Machine learning endeavors to adapt the way humans learn to computing. Consider the task of classifying several pictures or either cats or dogs. A human can complete this task by expending virtually zero effort because of their prior experience. Even children can readily differentiate between a cat and a dog. Now assume that the human had never seen a cat or a dog. If we wished to have a person complete this classification task, we would have to train them first by providing several example pictures labeled with their class membership (cat or dog). The person, through their training, could ascertain a set of features that can be used to discriminate between cats and dogs. Features such as size, shape, eye shape, color, and et cetera. The trained person could then classify previously unseen pictures as being either class or dog. Rather than algorithmically analyzing pictures of cats or dogs for relevant features, people develop experience and the process is automatized.

The approach taken by machine learning is analogous, but substantially different. Instead of being able to recognize pictures as either being cat or dog by experience, machines calculate the probability that a picture is a cat or a dog given the features. The model presumably has certain parameters like mean and standard deviation. A loss function provides a measure of the error incurred as a function of the parameters only, for given data, as it classifies labeled data. A machine learns by choosing the parameters that minimize the errors that it makes as it classifies (for example).

The classification task is schematically the same for the person and the machine in the sense that after some training they/it should be able to classify pictures they/it haven't seen. We can think of machine learning as a set of computer algorithms that allow computers to learn without explicit instructions. Although machine learning is used for tasks ranging from chat bots to the self-driving car, we will restrict our discussion to algorithms used to classify data. The product of learning in this context refers to the ability of a trained (learned) model to classify data that it has not previously seen. Learning refers to the calculation of the parameters of a model by minimization of a loss function with respect to the parameters of the model (θ). Minimization is carried out algorithmically, via gradient descent, rather than analytically.

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha^{(t)} \nabla J(\boldsymbol{\theta}^{(t)}) \tag{2.3}$$

where $\boldsymbol{\theta}^{(t+1)}$ is the "updated" set of parameters, $\boldsymbol{\theta}^{(t)}$ is the value of the parameters before update, $\alpha^{(t)}$ is the learning rate for the t^{th} iteration, and $\nabla J(\boldsymbol{\theta}^{(t)})$ is the gradient of the loss function evaluated at $\boldsymbol{\theta}^{(t)}$.

The gradient descent algorithm is a convex optimization algorithm that finds the optimal $\boldsymbol{\theta}^*$ given an initial guess $\boldsymbol{\theta}^0$ by moving in the direction of the gradient of the loss function, which is the direction of steepest descent towards the minimum[32].

Provided the loss function is convex, then the gradient descent algortithm is guarenteed to find a local minimum for appropriate choice of $\alpha^{(t)}$. If $\alpha^{(t)}$ is too large, then the search algorithm oscillates around a minimum. If it is too small, then the algorithm takes too long to run. The optimal step size is proportional to the slope of the function, so that the search

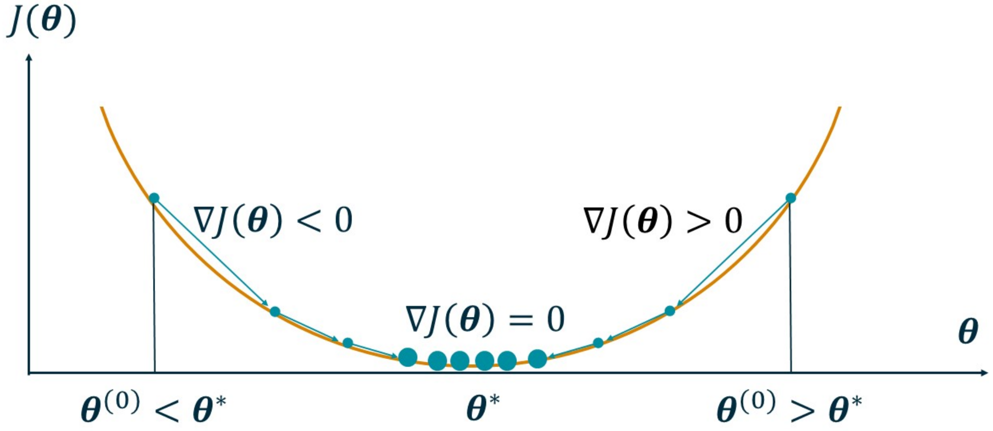

$J(\boldsymbol{\theta})$
$\nabla J(\boldsymbol{\theta}) < 0$
$\nabla J(\boldsymbol{\theta}) > 0$
$\nabla J(\boldsymbol{\theta}) = 0$
$\boldsymbol{\theta}$
$\boldsymbol{\theta}^{(0)} < \boldsymbol{\theta}^*$
$\boldsymbol{\theta}^*$
$\boldsymbol{\theta}^{(0)} > \boldsymbol{\theta}^*$

Figure 2.5. The figure shows two cases for the gradient descent algorithm, **1)** where the initial guess $\boldsymbol{\theta}^0$ is less than the optimal point $\boldsymbol{\theta}^*$, in which case $\nabla J(\boldsymbol{\theta}) < 0$ and by eqn 2.3 $\boldsymbol{\theta}^{(t+1)} > \boldsymbol{\theta}^{(t)}$ **2)** the initial guess $\boldsymbol{\theta}^0$ is greater than the optimal point $\boldsymbol{\theta}^*$, in which case $\nabla J(\boldsymbol{\theta}) < 0$ and by eqn 2.3 $\boldsymbol{\theta}^{(t+1)} < \boldsymbol{\theta}^{(t)}$ The weights are updated in amounts that are proportional to the gradient.

algorithm moves in smaller steps near a minimum [33]. It should be mentioned that machine learning is a very broad field such that the procedures that I have outlined only apply to a particular class of models called discriminative models, where we try to find lines (in 2-D) or hyper-planes (in many dimensions) to separate the data such that all of the data points belonging to one class are on one side of the line/plane and all of the data belonging to the other class is on the other side (Fig 2.6).

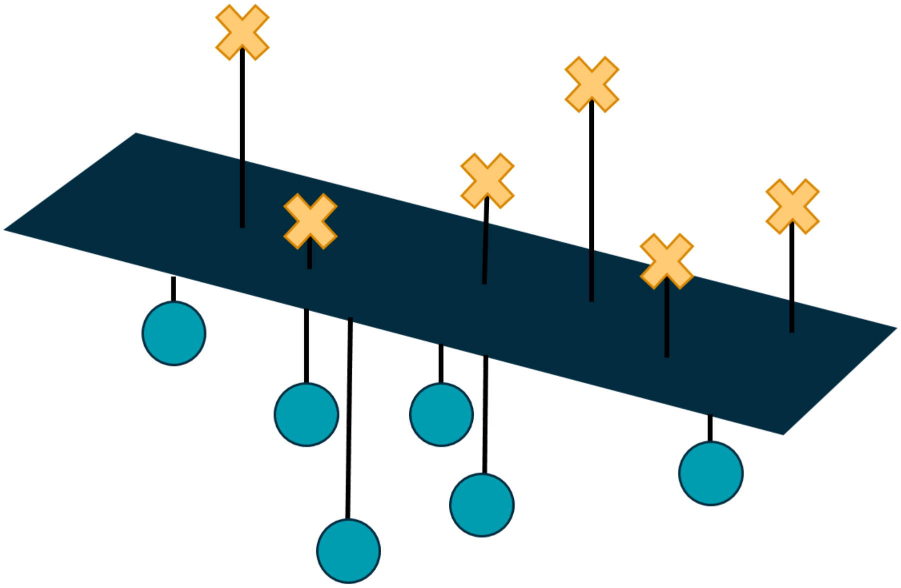


Figure 2.6. The separating hyperplane shown for the 2-class case. The optimal hyperplane minimized the residue between the data and the plane [33].

The result of implementing a machine learning algorithm is a function $y(x_\mathrm{i})$ that maps data points x_i to a real number representing a Bernoulli categorical variable (for the two class case). The precise form of $y(x)$ is determined during the training phase, by using the training set. Once the model is trained, it can be used to predict the class of new samples belonging to a testing set [32].

2.8.1 Machine Learning Experiments

In supervised learning, we assume that we have a pair of variables $(\boldsymbol{x}_{\mathrm{i}}, y_{\mathrm{i}})$, where $\boldsymbol{x}_{\mathrm{i}}$ is a d-dimensional input vector. The dimensionality of the input space is the number of features that are contained in the data. The variable y_{i} is a categorical variable corresponding to the true class of the i^{th} data point.

A *training* set is labelled data that is used to train a machine learning model. The trained model then makes predictions (for example) on the *testing* set. The model should not have been previously exposed to this data, for that would make it impossible to determine the ability of your model to make predictions on new data (generalization).

There are situations where you are given a dedicated training set and a testing set. For example, assume we want to build software that utilizes machine learning in real-time. Further suppose that we gathered some initial data to build the machine learning model, this would be our training set. Our goal is to ensure that our model will be able to make valid predictions on data that will see during deployment, which would be our testing set.

Another situation is where you have a single set of data. For example, data that you obtained from a single source (like the University Registrar). In this case, the data is split into a training set, and a testing set [33]. Still, the model should not be exposed to the testing set during training.

The testing set also should not be used for parameter tuning. In other words, machine learning algorithms have tunable knobs that are set by the user. The machine learning practitioner should not use the testing set to determine the optimal parameter settings for performance on the testing set. Instead, parameter tuning should happen during *validation.*

Model validation is where a portion of the training data is used as a mock testing set to determine the performance of the model on unseen data. This usually happens in a feedback loop where the training set is broken into a modified training set and validation set. The model is trained on the modified training set and tested on the validation set. The parameters of the model are adjusted, and the procedure is repeated until performance is saturated.

There are different methods for choosing how to partition you training data for validation.

The most common method is k-fold cross validation [34], where the training data is split into k partitions. For each of k trials, k-1 of the partitions are used for model training, and the last partition is used for testing (validation). This process continues until all partitions have been used in training and testing (fig 2.7).

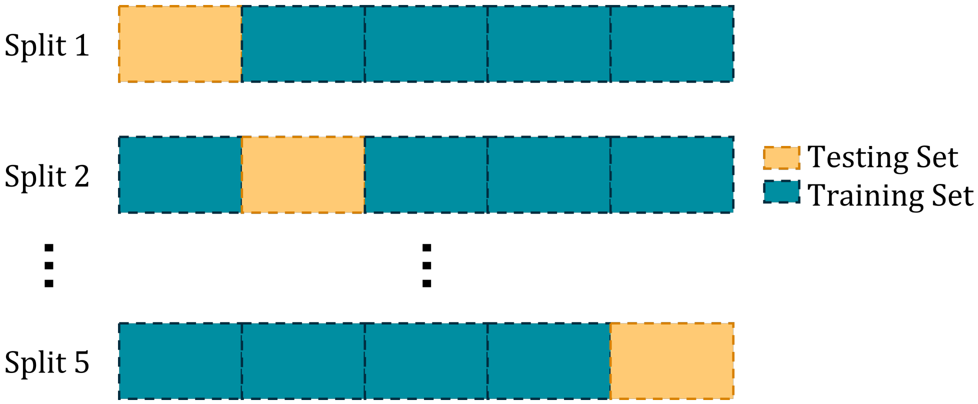


Figure 2.7. Figure shows k-fold cross validation for the case k=5

2.9 Linear Regression

Linear Regression is a regressive discriminative model whereby we seek to obtain a linear discriminant function of the form:

$$y_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \tag{2.4}$$

by using a training set to find the optimal value $\boldsymbol{\theta}^* = (\mathbf{w}^*, w_0^*)$. $\mathbf{w}$ is the weight vector which scales the importance of each feature in our data. $\mathbf{w}$ plays the role of the slope for a linear discriminant and w_0 is the bias, which plays the role of a y-intercept.

Linear regression is characterized by the mean-square error loss function.

$$J(\theta) = \frac{1}{n}\sum_{\mathrm{i}=1}^{n}(\mathbf{w}^T\mathbf{x}_{\mathrm{i}} + w_0 - y_{\mathrm{i}})^2 \tag{2.5}$$

which is exactly equal to the vectorized version

$$J(\theta) = \frac{1}{n}\left(\boldsymbol{X\theta} - \boldsymbol{y}\right)^T\left(\boldsymbol{X\theta} - \boldsymbol{y}\right) \tag{2.6}$$

where:

$$\boldsymbol{X} = \begin{bmatrix} | & | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \dots & \boldsymbol{x}_N \\ | & | & | & & | \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{2.7}$$

and the row of ones at the bottom is to account for the fact that $\boldsymbol{\theta}$ contains the weight vector $\boldsymbol{w}$ and the bias term w_0.

The discriminant function represents a hyperplane in a d-dimensional space, where d is the number of features. The optimal discriminant function is a separating hyperplane found by minimization of $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. While gradient descent would certainly yield a global minimum for such a simple loss function, and gradient descent is sometimes used in practice for large datasets, we can analytically obtain the optimal $\boldsymbol{\theta}$ [33]. We wish to solve:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0 \tag{2.8}$$

Plugging in $J(\boldsymbol{\theta})$

$$\nabla_{\boldsymbol{\theta}} \left(\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\right)^T \left(\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}\right) = 0 \tag{2.9}$$

Expanding

$$\nabla_{\boldsymbol{\theta}} \Big((\boldsymbol{\theta}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{\theta} - 2\boldsymbol{y}^T \boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{y}^T \boldsymbol{y} \Big) = 0 \tag{2.10}$$

Applying $\nabla_{\boldsymbol{\theta}}$

$$\nabla_{\boldsymbol{\theta}}(\boldsymbol{\theta}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}}(2\boldsymbol{y}^T \boldsymbol{X}\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}(\boldsymbol{y}^T \boldsymbol{y}) = 0 \tag{2.11}$$

Using $\frac{\partial}{\partial \boldsymbol{\theta}} \left[\boldsymbol{\theta}^T A \boldsymbol{\theta}\right] = A\boldsymbol{\theta} + A^T \boldsymbol{\theta}$ and $A = (\boldsymbol{X}^T \boldsymbol{X}) = (\boldsymbol{X}^T \boldsymbol{X})^T = A^T$

$$2(\boldsymbol{X}^T \boldsymbol{X})\boldsymbol{\theta} = 2\boldsymbol{X}^T \boldsymbol{y} \tag{2.12}$$

Finally,

$$\boldsymbol{\theta}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{y} \tag{2.13}$$

2.10 Naïve Bayes

Naïve Bayes is a generative discriminative model. Naïve Bayes is based on Bayes' Theorem, where we wish to find the conditional probability that a datum $\mathbf{x_i}$ belongs to a class C_k, $P(C_k|\mathbf{x_i})$.

By Bayes' Theorem,

$$P(C_k|\mathbf{x_i}) = \frac{P(\mathbf{x_i}|C_k)p(C_k)}{p(\mathbf{x})} \tag{2.14}$$

In Naïve Bayes, the posterior is assumed to be Gaussian. We find the probability of obtaining the dimension sample vector $\boldsymbol{x}_{\mathrm{i}}$ given class C_k as:

$$P(\boldsymbol{x}_{\mathrm{i}}|C_k) = \frac{1}{(2\pi)^{d/2}}\prod_{k=1}^{d}\frac{1}{\sigma_k}exp\left(-\frac{(x_{\mathrm{i}k}-\mu_k)^2}{2{\sigma_k}^2}\right) \tag{2.15}$$

This can also be written in a vectorized version,

$$P(\mathbf{x_i}|C_k) = \frac{1}{\sqrt{(2\pi)^d|\sum_k|}}exp\{(\boldsymbol{x}_{\mathrm{i}}-\boldsymbol{\mu_k})^T{\sum}_k^{-1}(\boldsymbol{x}_{\mathrm{i}}-\boldsymbol{\mu_k})\} \tag{2.16}$$

$\boldsymbol{\mu_k}$ is the mean vector for class k and $\boldsymbol{\Sigma}_k$ is the covariance matrix for class k.

This model is generative in the sense that we assume a Gaussian posterior probability distribution and use the data to calculate the parameters of the model The classification made for each datum is given by:

$$y = argmax_{C_k}\{P(\mathbf{x_i}|C_k)p(C_k)\} \tag{2.17}$$

The algorithm makes a classification by maximizing the posterior probability [33]. Implicitly, the use of Naïve Bayes assumes that each sample is independent from the rest and the features of the model are normally distributed with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Furthermore, we assume that the features of the model are conditionally independent, so that

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \sigma_d \end{bmatrix} \tag{2.18}$$

The assumption of independent and identically distributed data seems very restrictive, but this model works well for a number of applications including text classification in spite of violating this assuption [35].

2.11 Support Vector Machine

The support vector machine (SVM) is a classification scheme that much like linear regression is an applied optimization problem. The basic idea of SVM is that we wish, given two linearly-separable classes, to find a separating hyperplane that maximizes the distance between the hyperplane and the nearest data point (margin). The motivation is to create a classifier that is robust to noise. By making the margin as large as possible small perturbations (like noise associated with observed data) should not move the data across the hyperplane changing the classification, and thus improves generalization to unseen data [33], [36] (fig 2.8). Careful analysis shows that maximizing the margin is equivalent to the constrained optimization problem:

$$\text{argmin}_{\boldsymbol{w},w_0} \frac{1}{2} \parallel \mathbf{w} \parallel^2 \text{ subject to } y_{\text{j}}(\mathbf{w}^T\mathbf{x}_{\text{j}} + w_0) \geq 1 \tag{2.19}$$

The constraint $y_{\text{j}}(\mathbf{w}^T\mathbf{x}_{\text{j}} + w_0) \geq 1$ corresponds to a data point being correctly classified and laying some minimum distance from the hyperplane.

Using the weights and bias, $\boldsymbol{w}^*$ and w_0 obatined from the optimization problem (eqn 2.19) the discriminant is a line(plane)

$$y = \boldsymbol{w}^{*T}\boldsymbol{x} + {w_0}^* \tag{2.20}$$

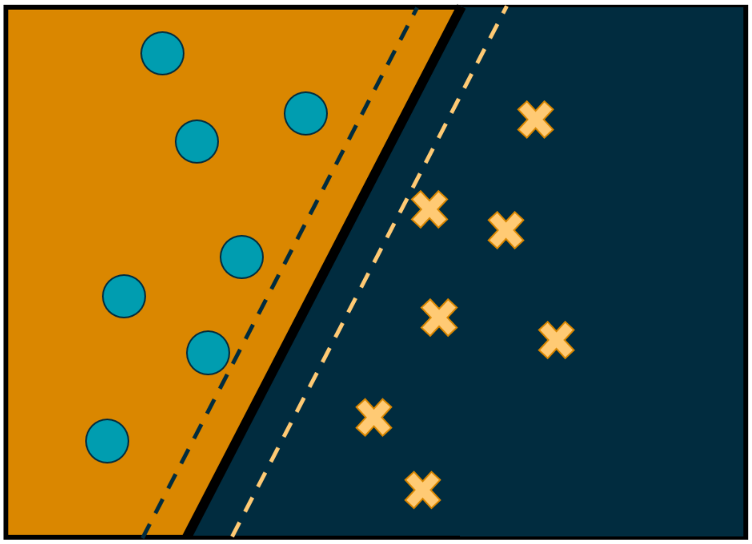


Figure 2.8. The separating hyperplane is shown as the solid black line dividing the blue and the orange regions and separating the two classes (blue circles, yellow x's). The margin is the distance from the hyperplane to the nearest data point [33]

2.12 Perceptron

The perceptron algorithm is a linear classifier that uses the hypothesis function

$$y = sign(\boldsymbol{w}^T\boldsymbol{x} + w_0) \tag{2.21}$$

Which is equivalent to

$$y = \begin{cases} +1 & \boldsymbol{w}^T\boldsymbol{x} + w_0 > 0 \\ -1 & else \end{cases} \tag{2.22}$$

The perceptron was created in 1950's as a way to emulate the function of a neuron. In other words, the perceptron can only take two values which is analogous to the two-state action potential of neuron [37].

The loss function of perceptron is:

$$J(\boldsymbol{\theta}) = -\sum_{\mathrm{j} \in M(\boldsymbol{\theta})} y_{\mathrm{j}}(\boldsymbol{w}^T\boldsymbol{x} + w_0) \tag{2.23}$$

Where $M(\boldsymbol{\theta}) := \{1, 2, \ldots, m\}$ is the set of mis-classified samples. We wish to find $\boldsymbol{w}$ and w_0 that minimize eqn 2.23.

2.13 Decision Tree

A decision (classification) tree is a binary-tree structure used infer an item's class based on some collection of features of the item. A decision tree originates on a root node (best predicting feature) and branches to a decision node (another feature) or a terminal node (class prediction) breaking a data set into subsets moving from child to parent within the network [38].

The tree is constructed feature wise, where an impurity measure is used to determine how predictive a given feature is. Impurity refers to the imperfect classification of data, given classification based only on that feature. There are several impurity measures, we will focus on the Gini impurity because it is used by default in *scikit-learn*'s implementation of decision trees (random forrests). The impurity is calculated for each division of the data and used to determine the optimal structure of the tree. The gini impurity in the case of a two-class problem is defined as:

$$I_{gini} = 1 - {P_1}^2 - {P_0}^2 \tag{2.24}$$

Where P_1 is the fraction (frequentist-probability) of data points that belong to class '1' given that it was assigned to a particular child node, and likewise P_0 for class '0'. Let's consider the following example. Assume we have the following data (table 2.1):

Table 2.1. Example data consisting of "HS GPA > 3.0", "Math ACT/SAT > 0.75", and "Took AP Phys" as features. "Passed The Course" is the label

N	HS GPA > 3.0	Math ACT/SAT > 0.75	Took AP Phys	Passed The Course
1	y	y	y	y
2	n	y	y	n
⋮	⋮	⋮	⋮	⋮
228	y	n	y	y

Consider the first column ”HS GPA > 3.0”. Data points are sorted on the basis of whether they satisfy this condition (fig 2.9).

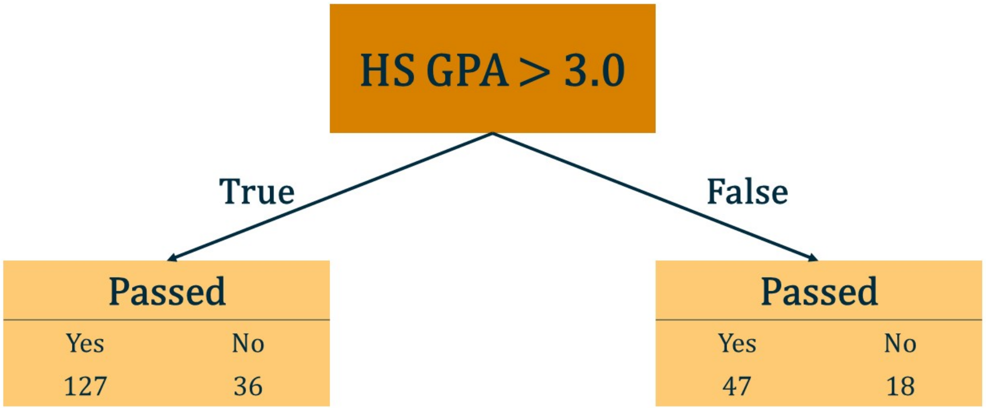


Figure 2.9. Data is sorted into child nodes based on whether they satisfy the condition ”HS GPA > 3.0”, and they are futher divided based on whether they passed the class (label).

Then the impurity is calculated for each of the child nodes in the following fashion:

For the ”True” branch:

$$I_{true} = 1 - \left(\frac{127}{127+36}\right)^2 - \left(\frac{36}{127+36}\right)^2 = 0.34 \tag{2.25}$$

For the ”False” Branch:

$$I_{false} = 1 - \left(\frac{47}{47+18}\right)^2 - \left(\frac{18}{47+18}\right)^2 = 0.40 \tag{2.26}$$

Then the total gini impurity is the average for each branch, weighted by the number of data points in each branch.

$$I_{total} = 0.71(I_{true}) + 0.29(I_{false}) = 0.71(0.34) + 0.28(0.40) = 0.35 \tag{2.27}$$

This procedure is repeated for all the features, and the one with the lowest impurity is used as the root node. At this point it should be mentioned that the tree will only split to include new nodes (features) if doing so reduces the gini impurity. Subsequently, this procedure is repeated for all of the remaining features until the left side of the tree is filled out. Finally, this procedure is repeated for each feature until the right side of the tree is built out. Following this example, a fully constructed decision tree with these features may look like fig. 2.10 (Note 80/20 [for example] means 80 people passed and 20 people failed)

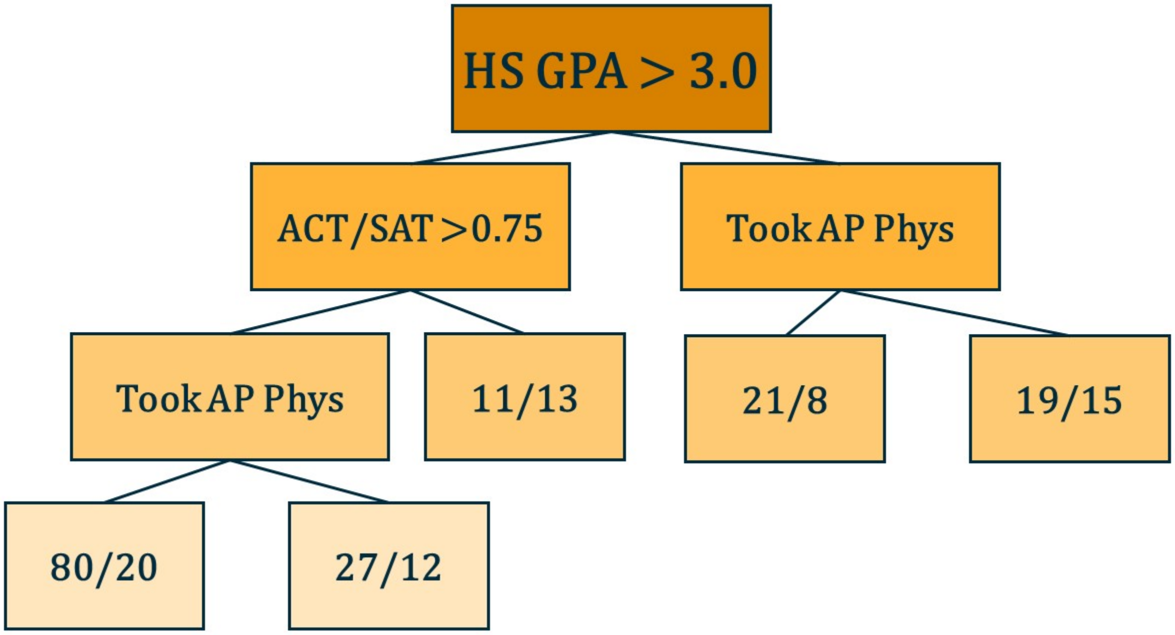


Figure 2.10. The figure shows a hypothetical decision treefully built from the data (table 2.1). The final classification is shown as N_{passed}/N_{failed}

Decision trees tend to be noisy and over fit to their training sets [34]. In the next section, we see how to deal with this limitation

2.14 Random Forest

A Random forest is a multitude of such decision trees where each item is classified by several decision trees. During training, the features for each tree are chosen from a random subset of the training data (with replacement), a process known as bootstrapping. The final classification for each datum is the mode of the classifications made by each tree [39], a process known as bagging. For a sufficiently large number of relevant features, random forests are robust to inclusion of noisy features [34]. This process of *bagging* and *bootstrapping* overcomes (to a large extent) the limitations of individual decision trees.

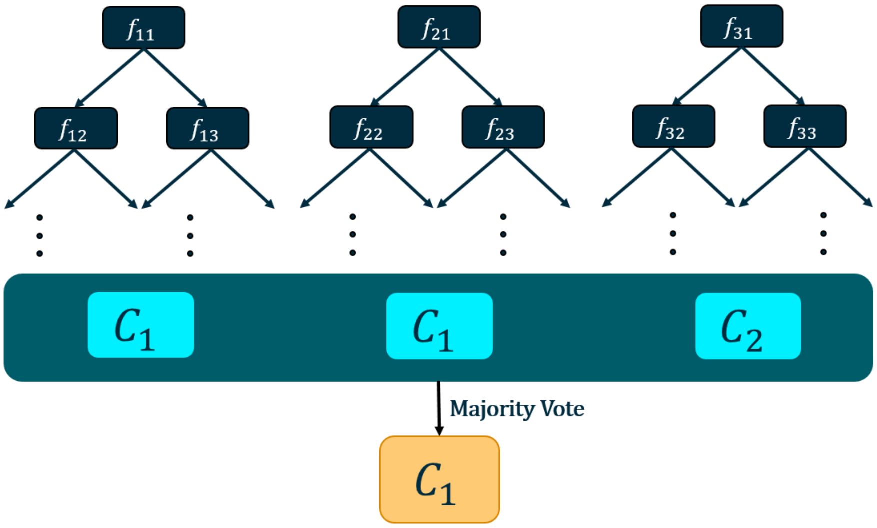


Figure 2.11. A random forest as a ”forest” of decision trees, where the overall classification is the mode of the classification from each tree.

2.15 Deep Learning/Multi-Layer Perceptrons

Neural networks (NN) are a parallelization of machine learning methods that is inspired by human models of cognition. The structure of a NN is decided at the time the model is constructed, usually based on the performance of the model on some validation data set. NNs consist of multiple layers of nodes connected sequentially (layer-wise). Data enters the network through the input layer. The features of the data are mapped to the nodes of the next layer by a series of weights indicating the strength of the connection between that

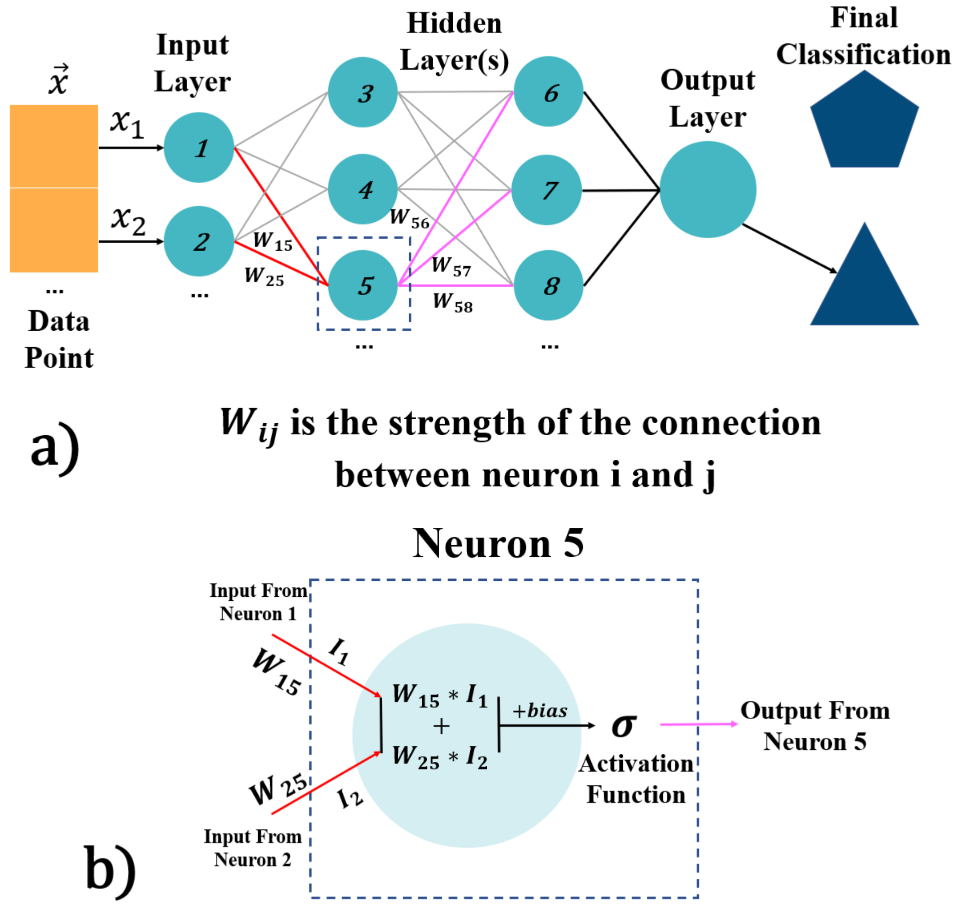


Figure 2.12. **a)** The structure of a neural network is shown where W_{ij} represents the strength of the connection between neuron i and neuron j. **b)** The input to neuron 5 from the previous layer I_1 and I_2 is multiplied by the weights, where W_{15} is the strength of connection between neuron 1 and neuron 5, and likewise W_{25} neurons 2 and 5. This setup is commonly referred to as a multi-layer perceptron

feature and the corresponding nodes (fig 2.12).

Just as electrical signals propagate in the brain by activation of neurons, the nodes in each layer run the output from the previous layer through a mathematical (activation) function, indicating the extent to which that neuron (node) is activated. In general, there are several of these layers known as hidden layers. In the output layer, the signal from the hidden layers is mapped to the output space to make a final prediction. Since the weights (strength of connection between consecutive layers) are randomly initialized, the first run isn't very accurate. Back propagation is used to recognize errors made by the network and update the weights throughout the network to improve predictive power. A thorough discussion of neural networks is beyond the scope of this report.

2.16 Natural Language Processing

Many educational applications of machine learning involve the use of text data. Since machine learning algorithms use statistics and can't directly understand language, we must find indirect methods to deal with this text data. To that end natural language processing can be used to perform computation and statistical inference with this text data.

Natural language processing (NLP) is a subfield of computer science in which a computer uses human language for computational tasks (e.g. classification). NLP began as algorithms based on explicit instructions [40]. Today, most NLP applications take a statistical approach and rely heavily on machine learning.

Common machine learning approaches to NLP applications such as spam filters, chat bots, recommender systems, and translation bots all require that words be transformed to numerical feature vectors. One common approach is a bag of words model (fig 2.13, which is a count vectorization method where a corpra (collection of documents, each called a corpus) is scanned for unique words and each corpus is transformed into a vector whose dimensionality is the number of unique words in the corpra and each component is the number of times that word appears in the corpus.

Another commonly used approach to word vectorization is the TFIDF (term frequency-inverse document frequency) weighting scheme (eqn 2.16).

$$W_{\mathrm{i,j}} = tf_{\mathrm{i,j}} log\left(\frac{N}{df_{\mathrm{i}}}\right) \tag{2.28}$$

$tf_{\mathrm{i,j}}$ is the frequency of term i, in document j, df_{i} is the number of documents containing term i, and N is the number of documents.

the TFIDF transformation is applied to every component of a count vector. TFIDF

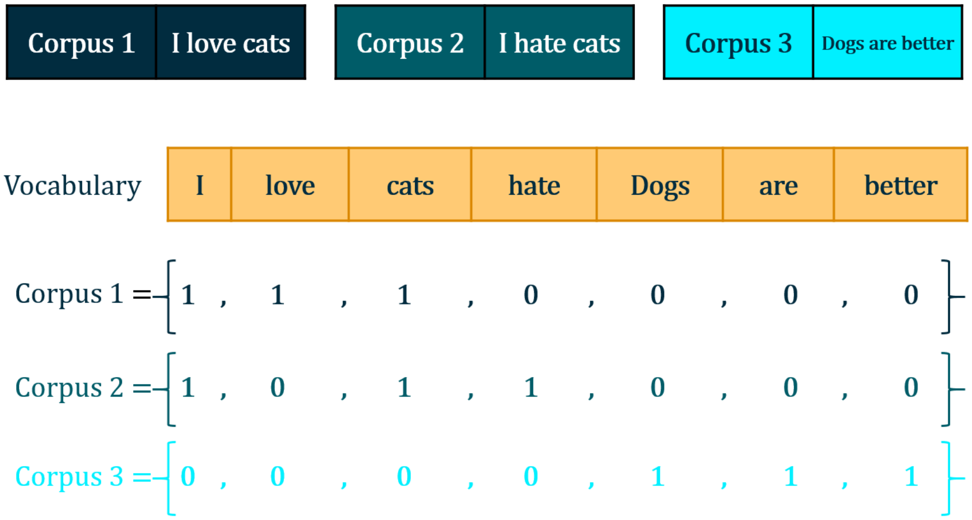

Corpus 1
I love cats
Corpus 2
I hate cats
Corpus 3
Dogs are better
Vocabulary
I
love
cats
hate
Dogs
are
better
Corpus 1 = [1 , 1 , 1 , 0 , 0 , 0 , 0]
Corpus 2 = [1 , 0 , 1 , 1 , 0 , 0 , 0]
Corpus 3 = [0 , 0 , 0 , 0 , 1 , 1 , 1]

Figure 2.13. Shows how the bag of words model (count vectorization) transorms words to vectors.

weighting linearly rewards for a term being common in a particular document, but punishes logarithmically for that term being common in all of the documents (fig 2.14).

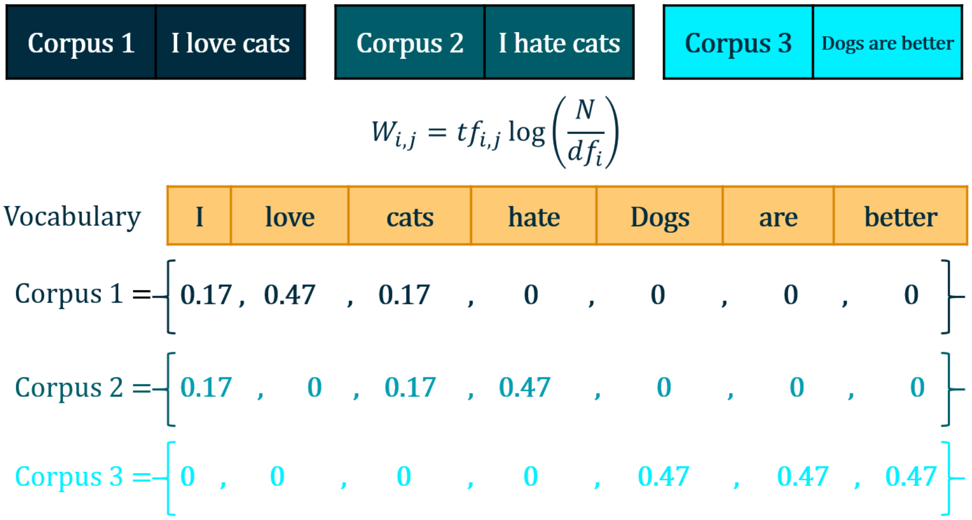

Corpus 1
I love cats
Corpus 2
I hate cats
Corpus 3
Dogs are better
$W_{i,j} = tf_{i,j} \log\left(\frac{N}{df_i}\right)$
Vocabulary
I
love
cats
hate
Dogs
are
better
Corpus 1 = [0.17 , 0.47 , 0.17 , 0 , 0 , 0 , 0]
Corpus 2 = [0.17 , 0 , 0.17 , 0.47 , 0 , 0 , 0]
Corpus 3 = [0 , 0 , 0 , 0 , 0.47 , 0.47 , 0.47]

Figure 2.14. Count vectors TFIDF transformed.

2.16.1 Word Embeddings

Word embeddings are a step-up in sophistication from word count encodings, in that capture the meaning of words in the following fashion. Word embeddings are vectors that encode each word's position in a semantic space. Words with similar meaning are neighbors in such a space in the sense that similar words are closer together than dissimilar words. For example, truck is closer to car than horse or mule (fig 2.15).

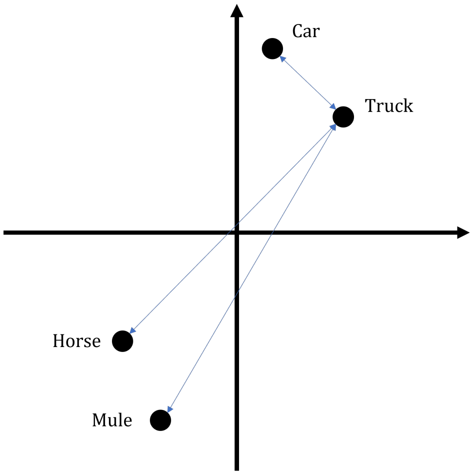


Figure 2.15. Words are represented vectors where meaning is inferred from where vectors live in this semantic space.

In practice, a set of word embeddings encodes the essay as a set of high dimensional (e.g. 300 dimensions) dense vectors. Each word in the essay is replaced by its embedding vector (fig 2.16). This operation isn't actually performed on each essay individually, it is performed as a matrix operation on the entire corpra. This produces a very complicated data structure called a tensor. For example, the shape of a tensor could be N x d x 300 where N is the number of essays, d is the number of words in each essay, and 300 is the dimensionality of the word embedding.

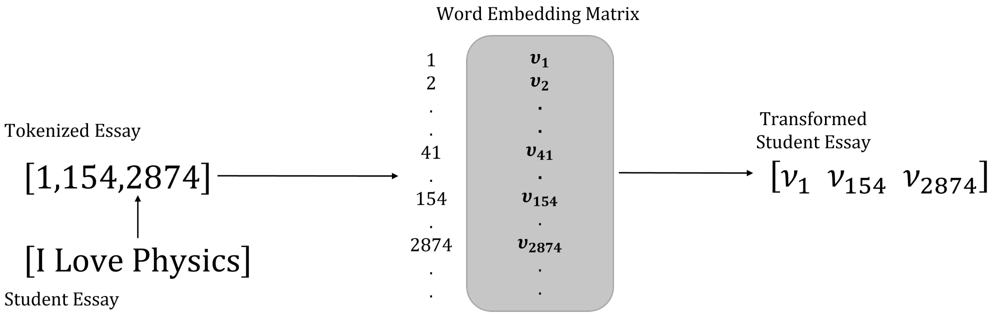


Figure 2.16. Shows the procedure of transforming essay to tensor of word embedding vectors

2.17 Prior Work on Adaptive Instruction and Machine Learning in Education

The Next Generation Science Standards (NGSS Lead States, 2013) and The Framework for K-12 education (NRC, 2012) emphasize learners using core ideas, scientific and engineering practices, and crosscutting concepts to make sense of phenomena and solve complex problems; to address learning goals that focus on transfer/application rather than recall [41]. One of the greatest challenges is the assessment goals specified in the K-12 framework and NGSS is to develop assessment tasks which tap complex constructs such as cognitive abilities during practice and to apply the assessment during instruction [42].

Complex, open ended assessment tasks are needed to accurately measure three-dimension learning [43], [44]. The scoring of three dimensional assessment tasks raises validity concerns [45]. Machine Learning (ML) has the capability to find patterns in data that offers a major advantage in automated scoring of student generated text in responding to tasks that require students to provide open ended arguments and explanations, construct diagrams, or describe scientific investigations [46]. Machine scoring has similar reliability to human scoring in English and Chinese [47].

A vast body of research has shown the effectiveness of machine learning (ML) in science assessments. Zhai, Yin, Pellegrino, Haudek,and Shi [48] have provided a systematic meta-analysis of the field. The authors analyzed the studies using a three-tier analytical framework, which examined the technical feature (i.e. advantage of automaticity), validity feature (i.e. empirical evidence and theoretical rationales supporting the inference), and the pedagogical feature (i.e. benefits of using ML in science assessments). Machine learning (ML) in science assessment uses one of three approaches: Supervised ML, Unsupervised ML, Semi-Supervised ML. In supervised ML, the machine learns from labelled data to develop an algorithmic model, and then infers and makes decisions using the trained model. Supervised ML has been used for automated scoring, such as assigning a score to students' scientific reasoning [49], [50]. Supervised ML has also been used for classification of student responses [51], recognition [52] and prediction of student performance, such as by analysing online student discussion to predict performance on a project [53]. In unsupervised ML, the model automatically performs the desired task using the latent structure of the data. Although supervised ML has great potential to automatically score student responses, this study finds that this approach requires a lot of data, which must be labelled at large cost, for training.

Unsupervised ML does not require labelled data to train the MLA. This reduces human effort and avoids high training costs. A large sample size is needed to increase validity. This requirement limits the efficiency of using ML in high-stakes testing, because complex assessment and grading tasks are usually time and cost consuming to grade during instructional practice [54]. Unsupervised ML typically uses two types of attributes: demographic, and academic. Muldner, Burleson, Van de Sande, and VanLehn [55] examined students who used an Intelligent Tutor System to make progress in an online tutorial system while not learning the underlying physics. Zehner, Saelzer, and Goldhammer [56] employed several unsupervised approaches to classify student responses to the Program for International Student Assessment (PISA). Unsupervised machine scoring can be used to analyze large qualitative data sets to reveal patterns in qualitative data [57]. For instance, ML has been used to code multimodal representational thinking in learners' written representation of lab reports. [58]. Among various techniques used, deep learning has been found to be more accurate in finding patterns than traditional methods. Finally, in Semi-Supervised ML part of the data set is labelled and part is not. Mason and Just[59] employed fMRI while presenting a set of 30 physics concepts one-by-one to students. They used a pixel-map of the brain scans to identify activations (labelled data) and combined this with unlabelled data to build a Naïve-Bayes model and predict, with 75% accuracy, whether the student understood the concepts.

Developing and using complex constructive response assessments in a classroom environment, whether K-12 or college, can provide teachers and students information to make educational decisions for student learning. This feedback must occur in a timely manner to be valuable to student learning (NRC, 2001). In addition to science assessment per se, several studies have also embedded ML in learning activities, usually using feedback or assistance provided by ML directly to students. In large college instructional contexts, ML can provide valuable instantaneous feedback [60], [61] to students and teachers on cognitively challenging tasks that go beyond selected response. For instance, ML has been used to automate scoring of constructed response assessment designed to elicit complex reasoning aligned to a physiology learning progression for undergraduate students [61], and various ML models have been explored on how they could be used to eliminate attrition among at-risk students [60]. ML has also been used to track learners' facial expressions to measure student engagement while performing investigations [62]. Further, ML-generated automated feedback when integrated seamlessly into online curriculum can influence student's performance, such as on students' response to different kinds of feedback on in computer simulation tasks [63]. Donnelly, Vitale, and Linn (2015) used ML to support scientific inquiry, by providing students with individual learning guides based on automated scoring of student essays. Gerard and Linn [64] found the combination of automated and teacher guidance was more effective for learning the topics of photosynthesis than automated guidance alone. Vitale, McBride, and Linn [65] found that specific learning guides showing students what was missing in their responses was less effective than those that provided hints or encouraged students to revisit their responses. Gerard, Kidron and Linn [66] explored how teachers customized automated guidance in their classrooms. Their results indicate that ML-based science assessment has the potential for adaptive learning and responsive teaching. However, it is difficult to promote the adoption and use of emerging technologies if they require excessive cost of time or money [67].

The works in the April 2021 special issue of the Journal of Science Education and Technology Krajcik [68] shows that ML can be used beyond scoring and providing general feedback to selective response items to students. Rather, ML can be used to reliably evaluate complex open-ended assessment tasks and provide almost immediate, or just in time feedback to researchers, students, and instructors on complex open-ended assessment tasks that show how learners use their knowledge. Immediate results allow teachers and instructors to tailor feedback to differentiate instruction to promote learning. Krajcik [68] argues that while this is a significant step forward and does support differentiation, the potential to provide more meaningful feedback to open-ended assessment items exists. For instance, feedback can be tailored to a student's response promoting the individual to deeper levels of understanding, and to identify less engaged learners [47]. Though it may be some time before the products of these works become available to students and teachers, the potential is staggering. The challenge of how to efficiently evaluate and provide quality feedback is one hurdle that educators around the globe will have to solve in order to produce educational systems that will enable students to use their knowledge to solve complex problems, make decisions, and learn more when needed. The assessment component of educational systems is a critical piece that ML can help to solve. ML also holds promise in helping college instructors in modifying instruction and materials for future uses. Overall, advances in ML are allowing science education researchers to analyze students' responses to complex assessment items, and improve assessment, instruction, and curriculum to better promote student learning.

2.18 Chapter Summary

In this chapter we've discussed cognitive load theory, dealing with the cognitive resources available to students and how they use them during learning and problem solving. We also reviewed some of the cognitive load effects that emerge from cognitive load theory. Specifically, we discussed the expertise reversal effect wherein educational materials that provide a high level of guidance and are effective for low prior knowledge students prove to be inferior for high prior knowledge students who perform better with less guided materials.

In a subsequent section we reviewed the features of adaptive learning platforms as well as some principles of design for adaptive learning systems. In my future work the expertise reversal effect will provide theoretical justification for the differences in the materials that we show to students classified as low prior knowledge and students classified as high prior knowledge.

In a following segment we had a discussion of the theory behind machine learning and mentioned some specific algorithms. Additionally, we talked about how we can use machine learning with text data by using the methods of natural language processing.

Machine learning/Natural language processing can be used in future productions of the modules to classify students by knowledge level, or even provide pre-emptive feedback to students.

3. PRIOR WORK

3.1 Pilot Study 1

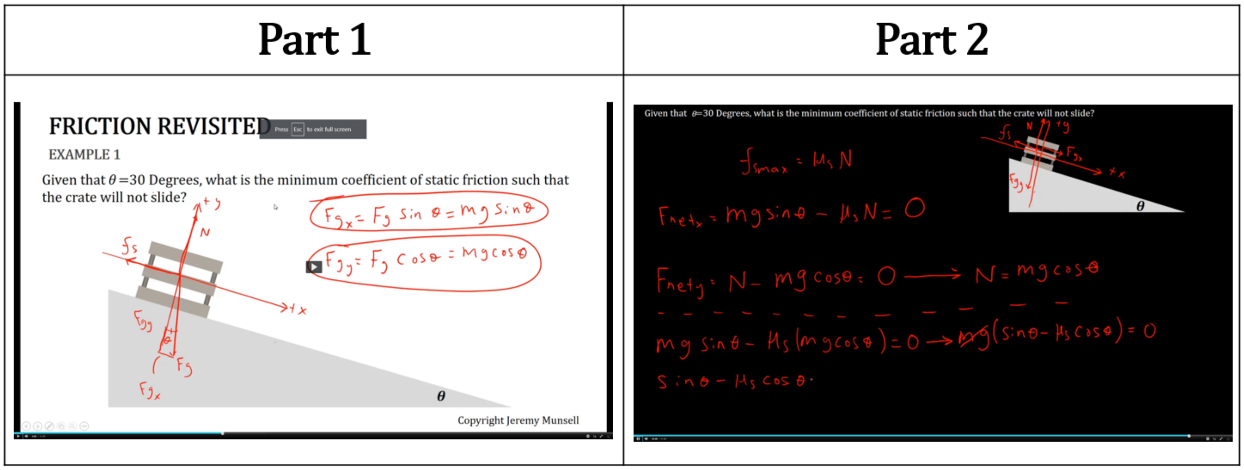
To what extent will students benefit from interactive online modules containing instructional videos that replace traditional classroom lectures and contain guided examples that scaffold expert-like problem solving process?

In pilot study 1 we tried to determine if students will benefit from interactive online modules containing instructional videos that replace traditional classroom lectures and contain guided examples that scaffold expert-like problem solving process

Physics 220 (Alg-based Physics 1) was offered as a summer course to N=28 students at Purdue University. This course used a flipped-classroom format where the lecturing was provided outside of class by way of interactive instruction modules and class time was used for summary of concepts and working examples. Additionally, weekly quizzes were used to supplement exams as an assessment tool to provide more timely feedback to students and cultivate a more equitable classroom [69]. The modules were required for 15% of the course total grade. Each module, which was created in Qualtrics, consisted of a training/lecturing phase where traditional lecturing was replaced by multiple short videos (about 5 minutes each) and a second phase where students were guided through a worked example. Each video contained lecture-style slides with animations and were recorded with voice narration. The training videos also included one or more worked examples, which were shown to aid schema acquisition [4], [5], [22].

The guided example phase of each module starts with a problem statement and is followed by questions intended to provide scaffolding to an expert-like problem solving process (Fig 3.2).

Students were given feedback product (correctness) and process (explanatory) feedback [22]. Students were also given process feedback even if they entered the correct answer, this was by their request (Fig 3.3). The Force Concept Inventory exam was administered on the first week and again at the completion of the mechanics portion.



Part 1

FRICTION REVISITED

EXAMPLE 1

Given that θ =30 Degrees, what is the minimum coefficient of static friction such that the crate will not slide?

$F_{g_x} = F_g \sin\theta = mg\sin\theta$

$F_{g_y} = F_g \cos\theta = mg\cos\theta$

Copyright Jeremy Munsell

Part 2

Given that θ=30 Degrees, what is the minimum coefficient of static friction such that the crate will not slide?

$f_{s_{max}} = \mu_s N$

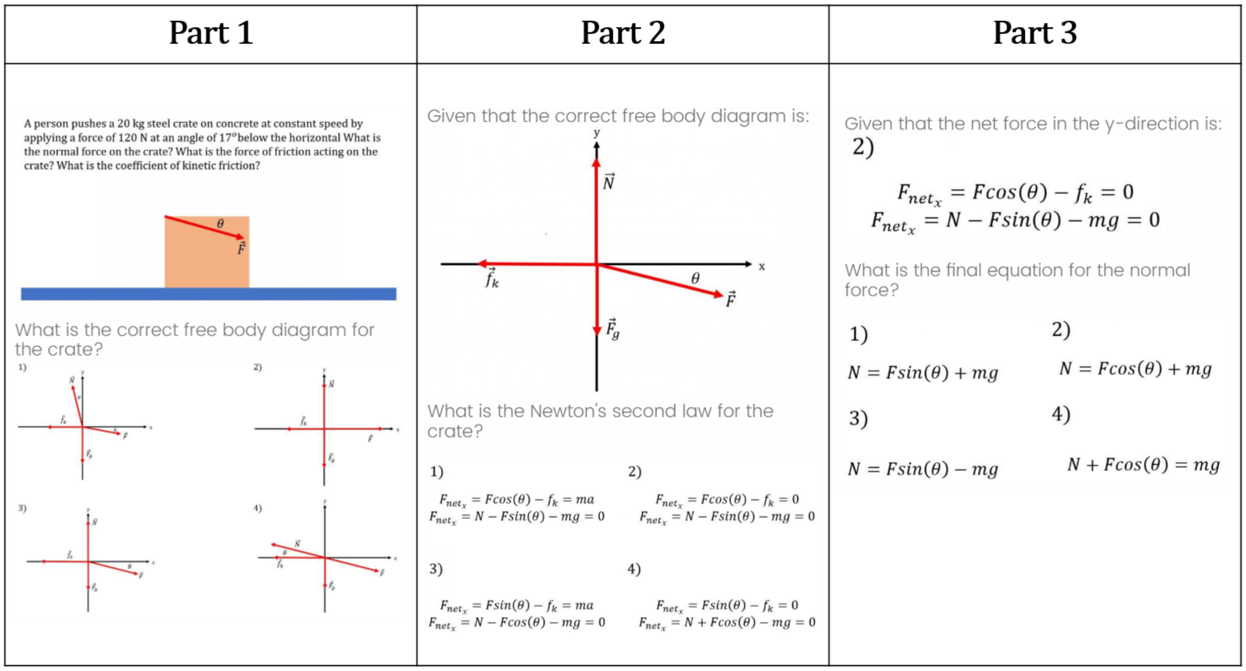
$F_{net_x} = mg\sin\theta - \mu_s N = 0$

$F_{net_y} = N - mg\cos\theta = 0 \longrightarrow N = mg\cos\theta$

$mg\sin\theta - \mu_s(mg\cos\theta) = 0 \rightarrow \cancel{mg}(\sin\theta - \mu_s\cos\theta) = 0$

$\sin\theta - \mu_s\cos\theta$.

Figure 3.1. Shows a worked example. The problem statement is shown on the left (part 1) and the solution shown on the right (part 2).



Part 1

A person pushes a 20 kg steel crate on concrete at constant speed by applying a force of 120 N at an angle of 17°below the horizontal What is the normal force on the crate? What is the force of friction acting on the crate? What is the coefficient of kinetic friction?

What is the correct free body diagram for the crate?

1)

2)

3)

4)

Part 2

Given that the correct free body diagram is:

What is the Newton's second law for the crate?

1)

$F_{net_x} = F\cos(\theta) - f_k = ma$
$F_{net_x} = N - F\sin(\theta) - mg = 0$

2)

$F_{net_x} = F\cos(\theta) - f_k = 0$
$F_{net_x} = N - F\sin(\theta) - mg = 0$

3)

$F_{net_x} = F\sin(\theta) - f_k = ma$
$F_{net_x} = N - F\cos(\theta) - mg = 0$

4)

$F_{net_x} = F\sin(\theta) - f_k = 0$
$F_{net_x} = N + F\cos(\theta) - mg = 0$

Part 3

Given that the net force in the y-direction is:

2)

$F_{net_x} = F\cos(\theta) - f_k = 0$
$F_{net_x} = N - F\sin(\theta) - mg = 0$

What is the final equation for the normal force?

1)

$N = F\sin(\theta) + mg$

2)

$N = F\cos(\theta) + mg$

3)

$N = F\sin(\theta) - mg$

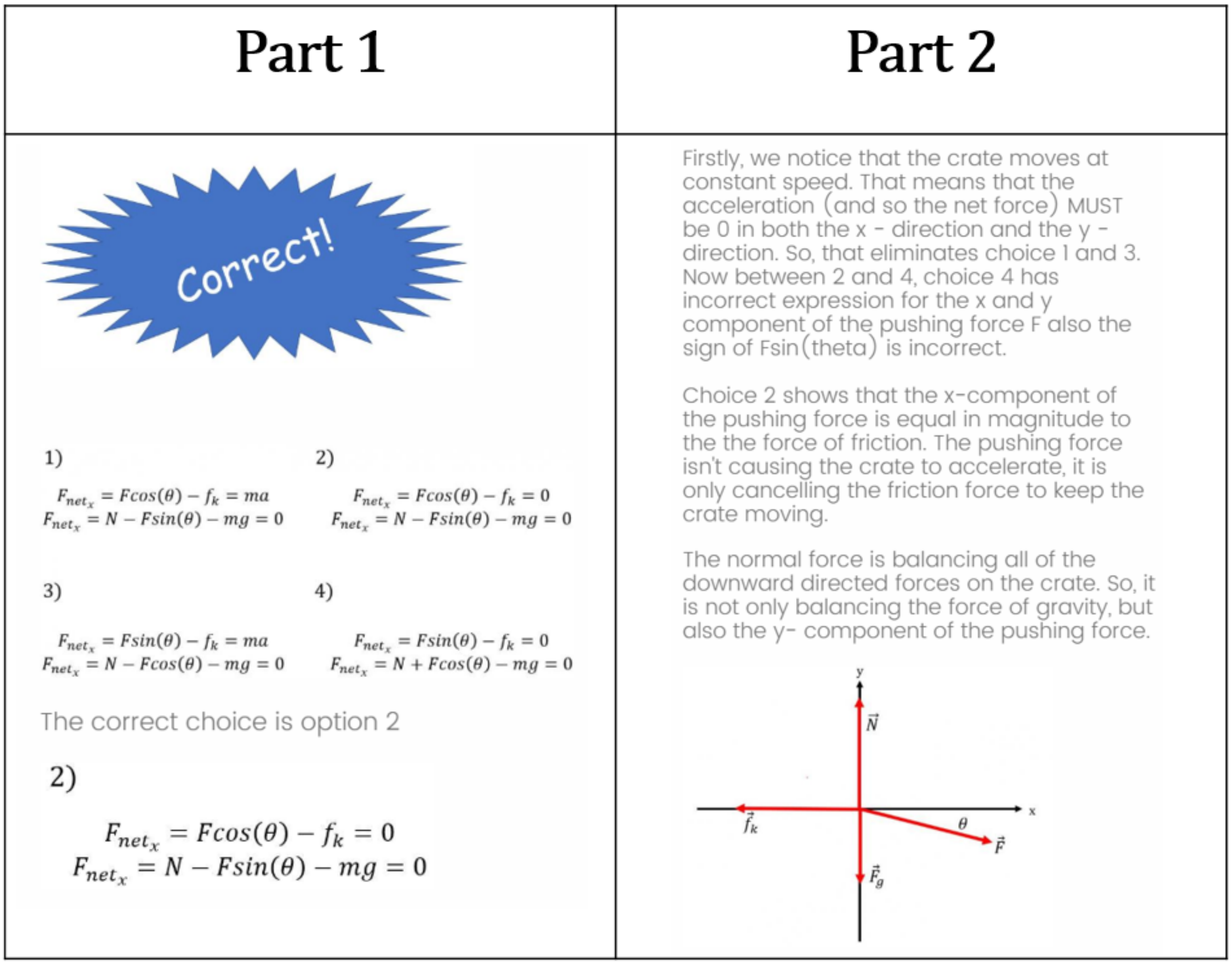
4)

$N + F\cos(\theta) = mg$

Figure 3.2. The figure shows the typical parts of a problem in a module (parts 1-3) correspond to the problem solving steps.

3.1.1 Results

The Force Concept Inventory exam (FCI) was administered in the first week of classes (pre-instruction) and again at the end of the mechanics section of the course (post-instruction).



Part 1

Part 2

1)

$F_{net_x} = Fcos(\theta) - f_k = ma$
$F_{net_x} = N - Fsin(\theta) - mg = 0$

2)

$F_{net_x} = Fcos(\theta) - f_k = 0$
$F_{net_x} = N - Fsin(\theta) - mg = 0$

3)

$F_{net_x} = Fsin(\theta) - f_k = ma$
$F_{net_x} = N - Fcos(\theta) - mg = 0$

4)

$F_{net_x} = Fsin(\theta) - f_k = 0$
$F_{net_x} = N + Fcos(\theta) - mg = 0$

The correct choice is option 2

2)

$F_{net_x} = Fcos(\theta) - f_k = 0$
$F_{net_x} = N - Fsin(\theta) - mg = 0$

Firstly, we notice that the crate moves at constant speed. That means that the acceleration (and so the net force) MUST be 0 in both the x - direction and the y - direction. So, that eliminates choice 1 and 3. Now between 2 and 4, choice 4 has incorrect expression for the x and y component of the pushing force F also the sign of Fsin(theta) is incorrect.

Choice 2 shows that the x-component of the pushing force is equal in magnitude to the the force of friction. The pushing force isn't causing the crate to accelerate, it is only cancelling the friction force to keep the crate moving.

The normal force is balancing all of the downward directed forces on the crate. So, it is not only balancing the force of gravity, but also the y- component of the pushing force.

Figure 3.3. The feedback shown consists of product feedback where ”correct!” is shown or it is not shown followed by process feedback.

The pre-instruction mean score on the FCI exam was 34.9% ± 2.7% and the post-instruction mean FCI score was 52.4% ± 3.0%.

This constitutes a normalized FCI gain (g) of 0.274 ± 0.05 with a statistically significant improvement between pre and post-instruction ($p < 0.00001$).

$$g = \frac{\langle post \rangle - \langle pre \rangle}{100 - \langle pre \rangle} \tag{3.1}$$

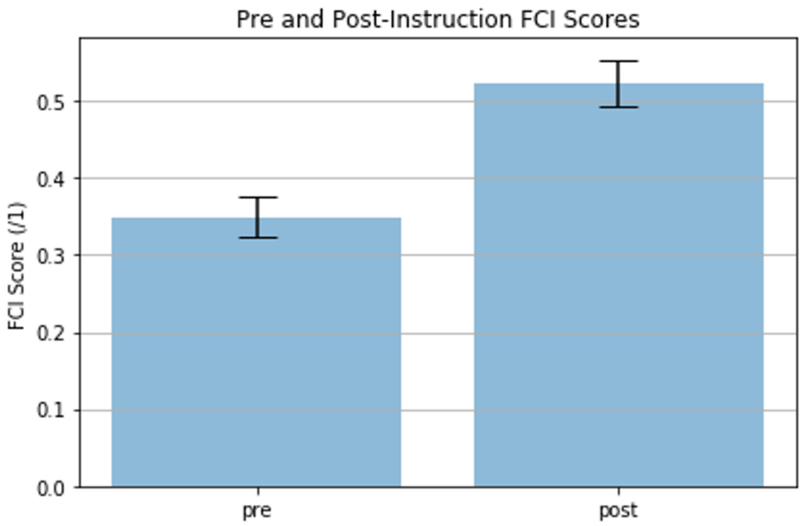


Figure 3.4. FCI pre and post scores in pilot study 1.

3.1.2 Discussion

The results of pilot study 1 indicate that the use of online instructional modules is effective in building mechanics domain knowledge as evidenced by the normalized FCI gain (eqn 3.1). The FCI is widely regarded as being a reliable tool in assessing conceptual knowledge about forces and motion. Since analysis over many populations shows no significant correlation between pre-instruction scores and normalized gain [70], it can reasonably be concluded that comparison of pre-instruction and post-instruction performance is a measure of the effectiveness of the instructional method of building knowledge of mechanics.

Student reactions to the use of the modules is another factor that should be considered. Overall, student experiences with the modules was very positive. In an anonymous survey of the class, 27/29 students (93%) indicated that modules helped them learn the material at their own pace. In a follow up survey given two weeks after the modules has been introduced (in a 7 week course) 20/22 students (91%) indicated that they would like to continue using the modules for the remainder of the course.

There were some issues with implementation that very likely affected the efficacy of the modules. The modules were being created as the course was in progress which made for a short duration between the assignment of the module and its due date.

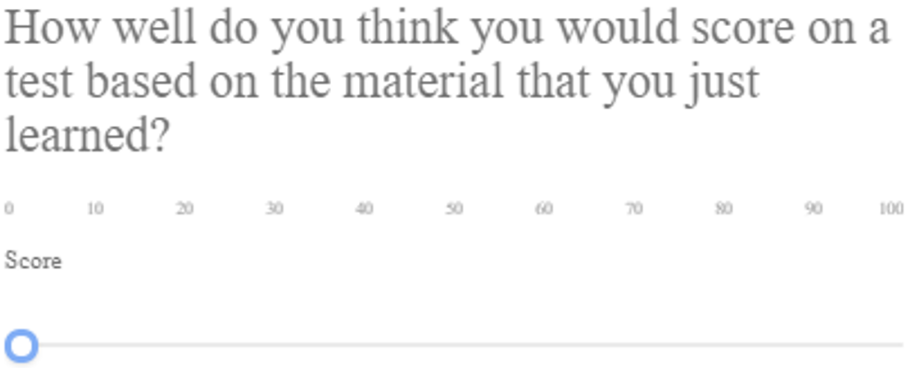
3.2 Pilot Study 2

To what extent can we make these modules adaptive to different knowledge levels optimize outcomes for students based on their prior knowledge?

The second study was a pilot study focused on structuring the modules from study 1 as an adaptive learning platform respecting students' level of expertise [22]. N = 798 students from a fall offering of Physics 220 were given the opportunity participate in this study in exchange for 2% of their total course grade. Four modules were created for selected topics including: Newton's second law, applying Newton's second law, circular motion, and energy. Each module was treated as independent in the sense that students could earn extra credit proportional to the number of modules that they completed. The modules were prepared at different levels. One level, designated as "high level guidance" commensurate with a high relative level of expertise. A second level, designated as "low level guidance" commensurate with a low relative level of expertise.

Students were shown a link on blackboard for each module. The link took them directly to a pre-assessment containing calculation questions relevant to the material contained in that module, designed to classify them according to their level of expertise. Half of the high level students were shown the high level guidance module and half were shown the low level guidance module and likewise for the low level students.

Following a training phase, the students were shown a judgement of learning question asking them to rate on a scale 1 – 100 how they felt they'd do on a test of this material based on their experience with the module.



How well do you think you would score on a test based on the material that you just learned?

Score

Figure 3.5. Judgement of Learning slider question.

Students were subsequently asked to complete a cognitive load survey asking them to rate their level of agreement, on a 9-point likert scale, with certain statements mapping on to the different kinds of cognitive load [71].

Table 3.1. Table shows the items from the cognitive load survey and their respective CL categories.

Intrinsic Load	The topic(s) covered in this activity was(were) very complex.
	The activity covered formulas that I perceived to be very complex.
	The activity covered concepts and definitions that I perceived to be very complex.
Extraneous Load	The intructions and/or explanations during the activity were very unclear.
	The instruction and/or explanations during the activity were, in terms of learning, very un-effective.
	The instruction and/or explanations during the acitivity were full of unclear language.
Germane Load	The activity really enhanced my understanding of the topic(s) covered.
	The activity really enhanced my understanding of physics.
	The activity really enhanced my understanding of the formulas covered.
	The activity really enhanced my understanding of concepts and definitions.

This survey was validated [71] as being sensitive as a measure of the different kinds of cognitive load experienced during instruction.

Finally students were given an assessment that was based on the initial assessment and was the same irrespective of the level of instruction received by the student. There were six questions in the final assessment. Two were conceptual questions following an "Answer/Reasoning" format, and the other four were near/far transfer questions based on the two initial calculation question.

A near transfer question is defined by us as using the same physical principles, and different context or different representation. A far transfer question uses the same physical principles with different context and different representation.

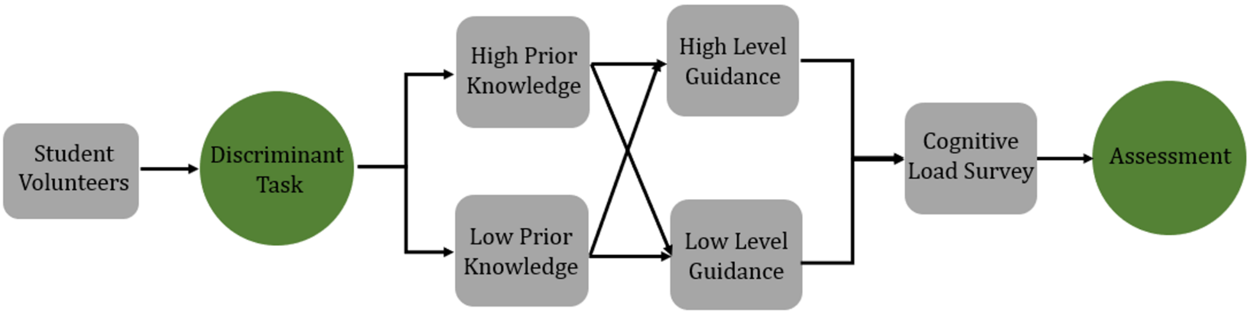


Figure 3.6. Structure of Pilot Study 2

3.2.1 Results

Students were designated as low prior knowledge (LPK) by a score of less than 130 on a 200 point mid-term exam ($\mu + 0.5\sigma$) and as high prior knowledge (HPK) for a score above (130). High and low prior knowledge students were randomly directed to a module with either a high (HLG) or low level guidance (LLG).

The assessment score for each condition was obtained by averaging over the assessment questions for each student, then averaging over all of the students in each condition.

In module 1 (on applying Newton's second law in 2-Dimensions) the HPK-HLG (High prior knowledge students shown a high level guidance module) significantly outperformed with other conditions ($p = 0.001$) on the assessment (Fig 3.7).

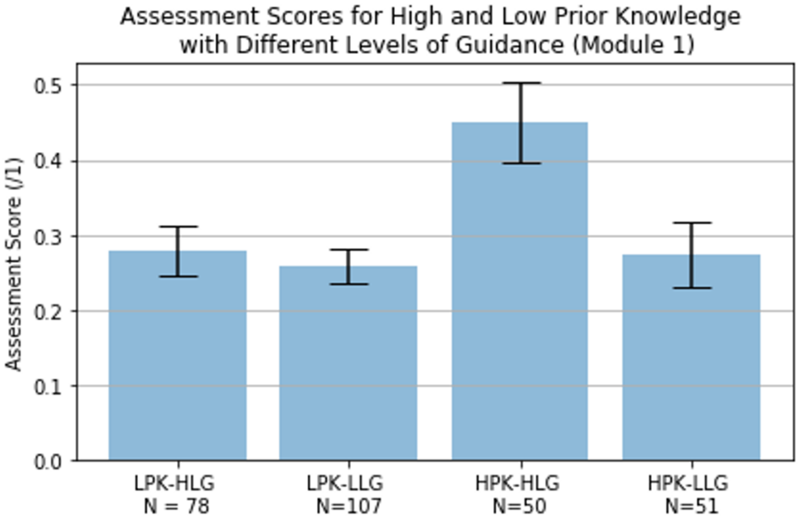


Figure 3.7. Assessment performance by condition for Module 1.

The cognitive load measures and judgements of learning reported by the participants are shown below (table 3.2).

Table 3.2. Cognitive load and judgement of learning for module 1.

Knowledge Level	Low Prior Knowledge		High Prior Knowledge	
Guidance Level	Low Level	High Level	Low Level	High Level
Intrinsic Load	5.8 ± 0.2	5.7 ± 0.02	5.8 ± 0.2	5.3 ± 0.3
Extraneous Load	4.1 ± 0.2	3.9 ± 0.2	4.0 ± 0.2	3.3 ± 0.3
Germane Load	6.4 ± 0.2	6.3 ± 0.2	6.0 ± 0.2	5.9 ± 0.3
Judgement of Learning	63.3 ± 2.0	68.9 ± 1.9	69.6 ± 2.5	73.8 ± 2.0

In module 2 (on circular motion), the HPK-HLG condition again showed a statistically significant higher performance ($p < 0.00001$) than the other conditions on the assessment questions (Fig 3.7).

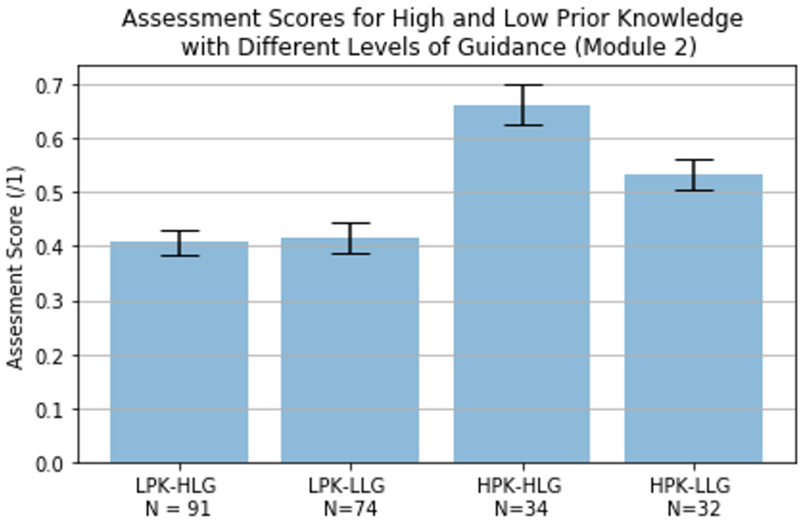


Figure 3.8. Performance on the assessment task for module 2 by conditions.

The self reported cognitive load measures and judgement of learning for each condition is shown below (table 3.3).

In Module 3 (on energy), the roles were reversed and the HPK-LLG condition showed

Table 3.3. Self reported cognitive load measures and judgement of learning.

Knowledge Level	Low Prior Knowledge		High Prior Knowledge	
Guidance Level	Low Level	High Level	Low Level	High Level
Intrinsic Load	5.8 ± 0.1	5.9 ± 0.02	5.1 ± 0.2	5.2 ± 0.4
Extraneous Load	3.2 ± 0.1	3.2 ± 0.2	3.1 ± 0.3	2.8 ± 0.3
Germane Load	6.2 ± 0.1	6.2 ± 0.2	5.8 ± 0.3	6.8 ± 0.3
Judgement of Learning	67.4 ± 1.5	68.9 ± 2.0	79.2 ± 1.7	80.1 ± 2.3

a statistically significant benefit ($p < .00001$) over the other conditions on the assessment task.

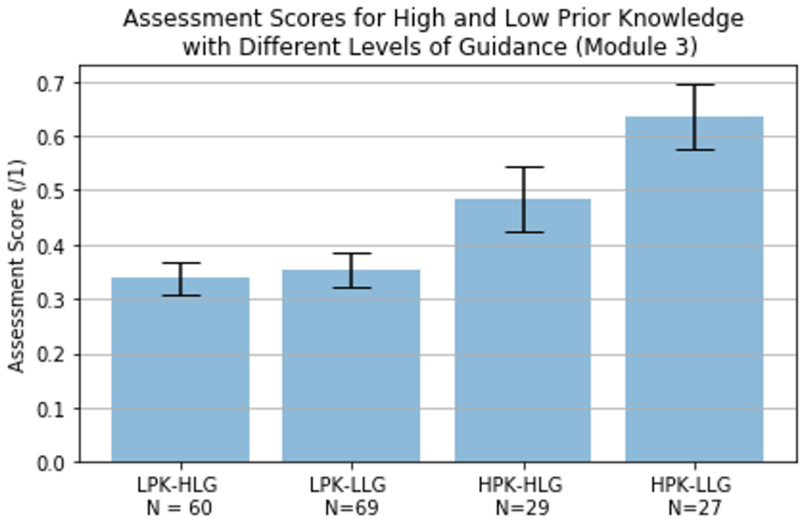


Figure 3.9. Performance of the assessment task for module 3 by conditions.

The self reported cognitive load measures and judgement of learning for module 3 is shown below (Fig 3.8).

3.2.2 Discussion

In the first two modules of this study, the lecture videos shown in the high/level guidance modules were the same. In the last module, the video included minimal guiding features

Table 3.4. Self reported cognitive load measures and judgement of learning score in module 3.

Knowledge Level	Low Prior Knowledge		High Prior Knowledge	
Guidance Level	Low Level	High Level	Low Level	High Level
Intrinsic Load	4.9 ± 0.2	5.0 ± 0.2	4.4 ± 0.3	4.3 ± 0.4
Extraneous Load	2.6 ± 0.2	2.6 ± 0.2	2.3 ± 0.2	2.3 ± 0.3
Germane Load	5.8 ± 0.2	6.8 ± 0.3	6.4 ± 0.3	6.8 ± 0.3
Judgement of Learning	62.0 ± 2.4	72.9 ± 1.9	80.7 ± 2.8	82.3 ± 3.2

and elaboration relative to the high level alternative. More work must be done on this project, but preliminary results suggest that the low level guidance module with minimally guided lecture videos show a greater benefit to high knowledge learners than the videos with higher levels of guidance. Regarding question 2(b) concerning how to classify students as high/low prior knowledge, this method of using the midterm exam score as selection criteria yields more sensible results. Classifying the students based on their performance on an initial task led to an outcome where there was no significant difference between the low/high prior knowledge conditions(Figure 20).

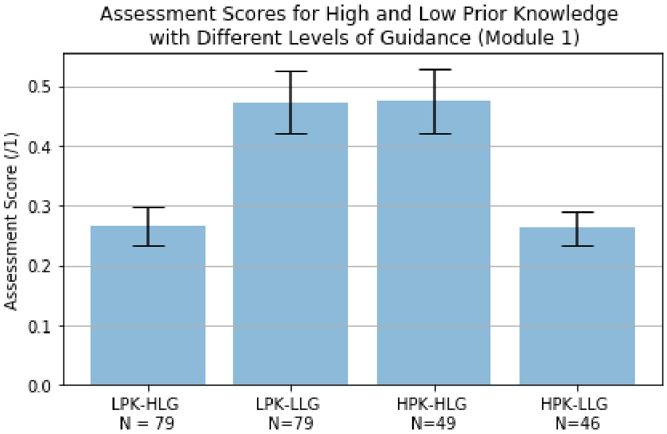


Figure 3.10. Performance of the assessment task for module 3 by conditions using an initial task for prior knowledge classification.

3.3 Pilot Study 3

Given a student essay outlining their strategy for solving problem, with what degree of accuracy can we predict who will ultimately answer the problem correctly using machine learning? In pilot study 3, N = 1289 first year engineering students enrolled in Physics 172 at Purdue university were given an online quiz where they were asked to solve a problem isomorphic to the ballistic pendulum. As the first step to their solution, they were asked to write a short essay elucidating their strategy to solve the problem, including the physical principles they would use and how they were going to use them. Their responses were collected on a spread sheet (.csv) along with their eventual outcome (correct =1 incorrect = 0) on the main problem. The goal of this study was to use a machine learning algorithm that would predict, given a student essay, if they'd go on to get the problem correct.

This work was executed in python using typical data science and machine learning libraries such as pandas, numpy, matplotlib, and scikit-learn.

First, the data consisting of student responses (essays) and class labels (correctness) along with irrelevant features (such as student ID) were imported to python using a pandas dataframe as an internal data-structure. The data was pre-processed by removing irrelevant columns (features) and dropping extra rows corresponding to multiple attempts, keeping the first attempt for each student.

Subsequently, the data was "cleaned" for the machine learning step. The data cleaning function performed the following steps:

- Removed all extraneous characters such as new line characters (\n) which are artifacts found in raw text, and html tags present when students used rich formatting in their online responses.
- Converted all the characters to lower case (since text methods are case sensitive).
- Irrelevant words (a.k.a. stop words) such as "is", "were", "and" were removed.
- Words replaced by their stems (energy, energetic, energetically → energi). This was performed for the sake of uniformity and also reduces the impact of spelling errors.

The data set was heavily biased towards people who got the problem correct (N = 935 in class 1, N = 354 in class 0) which can result in a lot of false positive errors. Therefore, a training set was constructed by randomly sampling 270 data (without replacement) from class 0 and 550 from class 1. The data set has a true proportion of 2.65 (class 1: class 0). The training set has a proportion of 2.03. This will hopefully make the model have a larger tendency towards false negative errors. The remaining 469 data were used as the testing set (N = 385 in class 1, N = 84 in class 0). It should be explicitly stated that the training set and the testing set are completely complementary.

In the next step, the corpra was vectorized. A count vectorizer (bag of words model) was fit on the entire data set. Then the training set and the testing were vectorized first by count vectorization then by tfidf weighting. The size of the feature space (unique 1200 words) was reduced by keeping the best 800 features chi-square correlated with the class label. This method was used to (hopefully) reduce the number of irrelevant, noisy features leading to better predictions.

The training set and the training labels were used to train a naïve Bayes classifier, a support vector machine, and a random forest classifier. The trained models are then used to classify the testing set.

3.3.1 Results

In machine learning tasks choosing an appropriate measure of error can be problematic. For example, the accuracy (which is defined as the number of correct classifications divided by the total number of classifications) can be very high if the classifier is largely biased towards the majority class, even though the classifier would likely have bad performance on an unseen example of the minority class. The confusion matrix is a compact way to express the error in a classification task by looking at the number of true positive classifications (They got the problem right and were classified as having got the problem right), true negatives (They got the problem wrong and were classified as having got the problem wrong), false positives (They got the problem wrong and were classified as having got the problem right), and false negatives (They got the problem right and were classified as having got the problem wrong).

Table 3.5. Figure shows a confusion matrix. True positive predictions are shown top-left. False negatives are shown top right. False positives are shown bottom left. True negatives are shown bottom right.

$N = 714$	Predicted: Correct	Predicted: Incorrect	
Actual: Correct	576.2 ± 2.34	29.8 ± 2.34	$N_{\text{Act. Correct}} = 606$
Actual: Incorrect	96.6 ± 1.22	11.4 ± 1.22	$N_{\text{Act. Correct}} = 108$
	672.8 ± 3.56	41.2 ± 3.56	Accuracy = 82.3%

3.3.2 Discussion

If this method is to be used to provide pre-emptive feedback, then our error rate is totally unacceptable. A false positive error amounts to a student who is likely to get the problem incorrect being told that they are on the right track. Likewise, a false negative error would result in a student who would probably go on to get the right being told that their they should reconsider their approach. With this rate or error, a pre-emptive feedback structure driven by this machine learning algorithm would do more harm than good.

This classifier makes predictions that the student will get the problem right by an overwhelming margin (15:1). One of the obvious reasons that the classifier performs so poorly is that the data from the two classes is very similar. If we look at the most commonly used words in both classes, words like ‘energy’, ‘momentum’, and ‘principle’ are at the top of the list for both classes. This conclusion is consistent with the poor (non-)performance of the support vector machine (SVM) classifier. The SVM classifier works by finding a separating hyperplane that maximizes the separation between the two classes in the feature space of the problem. If the two classes are not separable, because the selected features aren’t sensitive to the difference in the two classed, then the classifier will not be able to find such a separating hyperplane. Furthermore, when I artificially induced a difference in the two classes by manually deleting the common words such as ‘energy’, ’momentum’, and ‘principle’ from one class but not the other the accuracy of the classifier was 100%. This is a very hard problem, and the level of sophistication of our approach is simply inadequate.

4. PREDICTING AT RISK STUDENTS USING DATA MINING AND MACHINE LEARNING

4.1 ABSTRACT

Machine learning (ML) has been widely used in education for a wide variety of tasks. ML has been employed for automated scoring of text data, providing real-time feedback, and several other applications to improve student learning. In this work we report on the use of ML to predict students' risk of adverse outcomes in a first year algebra-based physics course for non-science majors. We obtained academic and demographic data for students. This data also included student's final grade in the course. This data set required significant processing before it was able to be used for ML.

This data set included students' final grade in the course. Their final grade was used to make a class label, where students receiving 'C+' or below were labeled '1' for the high-risk category, and students receiving 'B-' or above were labeled '0' for low-risk. We used eXtreme Gradient Boost (XGBoost) classifier to predict students in the high risk category with 70% precision, and those in the low-risk category with 78% precision. The overall prediction accuracy was 75%. The model was found to have very high accuracy in predicting students that recieved 'D+' or below as high risk. The model had much lower accuracy for students on the margins, students recieving 'B' or 'C'.

4.2 Introduction

Our modern society is constantly confronted with issues of science and technology. In order to have a future generation that is equipped navigate these challenges it is essential that students learn to solve problems, digest information, and reason based on evidence. These skills are developed in a curriculum rich in science, technology, engineering, mathematics, and computer science (STEM/CS) [72]. Therefore, ensuring a workforce with a sufficient number of STEM/CS competent graduates is a national priority [73].

Research shows that exposure to STEM in high school does not necessarily translate to increased post-secondary STEM degree attainment [74]. This implies that colleges and universities have a critical role to play in guaranteeing people enter the workforce with STEM/CS competency.

From the individual perspective, STEM/CS jobs are higher paying, more stable, and have more openings [75]–[77]. Furthermore, STEM degrees lead to increased opportunity as STEM/CS degree holders are able to transfer their skills to a broader range of jobs, and they typically are paid more even when working in non-STEM/CS fields [77], [78]. Unfortunately, Women and minorities are underrepresented in these fields in terms of number and pay [75], [78]. For these reasons, it is important to identify at-risk students and intervene to reduce STEM/CS attrition at the earliest possible opportunity.

Zhai, Haudek, Shi, Nehm, and Urban-Lurain [48] assessed the use of machine learning in education using a three-fold analytical framework. The validity feature corresponds to quantifying and assessing the performance of ML models. The technical feature examines the technology used and investigates its accessibility to students and teachers. Finally, the pedagogical feature delves into the potential benefit of this work to the field of education. We will present our findings in a similar manner.

Zariskie et. al. [1] reported on work predicting at-risk students. The authors collected data through educational data mining for students taking a calculus based introductory physics sequence for future scientists and engineers. Students were categorized by their final grade in the course. Students receiving an 'A' or 'B' were designated as low risk, and students receiving a 'C' or below were designated as high risk. Data comprised of academic and demographic variables were used to predict the student's risk of unfavorable outcomes. The data were collected in two samples and aggregated, from which a random split was used for training the machine learning algorithm, and another split was used for testing. Their performance in predicting outcome/risk for Physics 1 students was 73% using only institutional variables, but improved to 80% by the fifth week of classes by the inclusion of in-class variables (homework, quizzes, and exams) using random forest classifier and 82% using logistic regression with only in-class variables.

Yang et. al. [2] focused on predicting the 'most' at-risk students. In this work, they categorized the high-risk students as those receiving a 'D' or below in a calculus based Physics 1 course. Defining the high risk students in this way presented a significant challenge from the machine learning perspective, because categories were high imbalanced. Their initial performance using only institutional variables was 50% accuracy at predicting high risk students, which improved to 68% at the end of week 8 using a mix of institutional and in-class variables. In this study, aggregate data were randomly split into training and testing sets as in [1]

In this report we detail our work using only institutional data on students' prior academic behaviors and demographic information, which are available prior to the start of a semester, to predict students' risk profile. Specifically, we use institutional data obtained from the registrar to train a machine learning algorithm to *categorize* students' final course grade in a first semester algebra-based physics course for non-science majors. A student receiving an 'A' or 'B' is deemed low-risk, and likewise for students earning 'C' or below and high risk as in [1]. Students taking the course on a pass/fail basis were also included in the study where pass ('P') was included in low-risk, and fail ('N') was high-risk.

This work seeks to extend the pioneering work by [1], [2] by applying these methods to a different population of students. Namely, non-science majors enrolled in algebra-based introductory physics. Also we used a whole course as the testing set, rather than using random splits of aggregated data. We believe that using the data in this manner is more representative of how such a prediction model would be used in deployment.

Both works [1], [2] showed that the performance of machine learning models can be greatly improved by the addition of in-class variables as the semester progresses. We demonstrated above that it is imperative to identify at-risk students at the earliest possible opportunity. From this perspective, false negative errors cause the most harm. False positive errors, while still potentially causing harm to students, will likely be minimal if intervention strategies are not too cumbersome. Meanwhile, missing the opportunity to intercede for students at risk of failure can have disastrous consequences. The ability to identify at risk students (and also those not at risk) will improve greatly during the first weeks of class [1], [2] which will mitigate any harm caused by miscategorizing low-risk students as high-risk.

We define a false positive error as predicting a student who is not at academic risk as being in the high risk group. Likewise, we define a false negative error as predicting a student who is at academic risk as not being at academic risk. Therefore, we seek to minimize false negative errors. In light of these definitions, the research questions we addressed were the following:

1. **Research Question 1 (RQ1):** How should we define the high and the low risk groups such that the number of false negative errors are minimized?
2. **Research Question 2 (RQ2):** With what accuracy can we predict students that are at risk of failure?
3. **Research Question 3 (RQ3):** How can this work be used to benefit students?

In the methods section we will describe our data set and how it was processed. We will also describe our machine learning pipeline and explain how the experiment was conducted. In the results section we will state our results. Finally, in the discussion section we will evaluate our results in the conext of our research questions, and provide some concluding remarks.

4.3 Methods and Materials

In this section we discuss our data set, how features from the data set were processed, how relevant features were selected, and outline our ML pipeline.

4.3.1 Data

Our data set was acquired in two batches from the university registrar subsequent to IRB approval. The batches were prepared by two different analysts, by retrieving the information that we requested from a database. The first batch consisted of *approximately* 60 variables for $N_1 = 1705$ students from a first-year algebra- based mechanics course at a large midwestern university. Students from batch 1 were distributed across three semesters(Fall 2019 - Fall 2020) each taught by different instructors using a different modality (Completely in-person, completely online, and a hybrid [respectively]). Batch 2 consisted of *approximately* 60 variables for $N_2 = 676$ students. Students from batch 2 were from an in-person class that also had an online section taught during the spring 2021 semester. This course is mainly taken by students from the school of technology, pre-medical students, and students from the school of pharmacy. The academic risk profile of this population is higher than the corresponding calculus based intro-mechanics course for future scientists and engineers because these students possess weaker academic preparation.

The data from each batch were presented as 7 smaller sets of tabular data (spread sheets) corresponding to information about students' high school (HS) academic record, advanced placement (AP) math/science classes taken in HS, their ACT/SAT scores, demographic information, and information about college academic performance. The data were anonymized by use of an abridged version of their student identification number.

One of the challenges in working with this data set was a large number of missing values in the data. Values could be missing for any number of reasons. For example, if the student transferred from another college/university, they may not have been required to provide HS transcripts or ACT/SAT scores. Additionally, information for students coming from abroad

Table 4.1. Shows the Semester and Course Modality of Each Data Batch

Batch	Semester	Modality
1	Fall 2019	In-Person
1	Spring 2020	Hybrid
1	Fall 2020	Online
2	Spring 2021	Online

may not have been available for a variety of reasons. Also, students may appear more than once in data if they changed their major or they have a double major. More information is not available to determine the exact cause of multiple listing for the same student for logistical reasons.

Missing values were filled by mean / mode imputation. In this method, you fill missing values in a particular column with the mean (for numerical features) or the mode (for categorical features). This is a *quick and dirty* way to deal with missing values. More sophisticated methods use deep learning to fill missing values by using adjacent values as predictors. For example, you could fill a student's ACT/SAT score using their high school GPA, their AP Math scores, and etc. Early explorations determined that this approach, which is very time consuming, did not improve predictive power. Therefore, we used the less precise mean/median imputation technique. The record containing college academic data also included the final course grade reported to the registrar. This grade would include any modifications to mid-semester or final grades made by the different instructors. This feature will ultimately be used to create a label. Students earning 'B-' or above were deemed low-risk and assigned '0'. Students earning 'C+' or below were assigned '1', indicating they were high risk.

4.3.2 Combining Data

Although the data request submitted to the registrar for both batches of data was identical, the two batches differed substantially. In order to use the data for machine learning it had be combined homogenously. In other words, the data had to have the same features with the same column names. Furthermore, the batches had the data grouped differently among the subsets/records.

The records were combined by software in Python. The software loaded the data as a pandas dataframe, selected relevant columns, and mapped each column name to a desired target column name. Additionally, students missing their HS GPA or most recent prior term college GPA were dropped from the data. The software also handled some rough preprocessing like converted numerical values from strings to floats.

Student ACT/SAT scores were scaled by the respective maximum test score. This was performed to account for the fact that some students took one test but not the other. Math portions of ACT/SAT scores were combined into a single feature, averaging if more than one score was listed. ACT/SAT non-math scores were combined into a single feature in the same manner as the ACT/SAT math scores.

The Python program ultimately constructed two identical batches of data in terms of data used and the column names. The first batch, as defined in section 4.2.1 will be used as the *training* set, and the second batch is used as the *testing* set. Batch 1 had been used for previous work and therefore was not suitable for use in the testing set.

4.3.3 Processing

In order to maintain good data hygiene as in section 2.8.1, data processing performed by the software processed the batches (training and testing) separately. Information was allowed to flow from the training set to the testing set in the sense that will be outlined in this section, but never from the testing set to the training set. That is, any processing methods that require parameters such as a mean value or a standard deviation, obtained these values from the training set.

Machine learning algorithms generally perform better if numerical values are scaled to values less than 1. For one reason, using unscaled feature values can cause the model to place undue importance on features with larger values. Numerical features, such as HS and college GPA, were Z-scaled.

$$Z = \frac{(x - \mu_{train})}{\sigma_{train}} \tag{4.1}$$

Where x is the value, μ_{train} is the training set column mean, and σ_{train} is the training set column standard deviation. Training set and testing set values were scaled according to the training set values.

The model also makes use of several catgorical variables. Categorical variables are variables such as gender where values correspond to belonging to a particular category. In the original data, these items had string values such as 'm' or 'f' in the case of gender. In order to admit these features to the model, they must be transformed into a number or a vector.

One common approach to this problem is ordinal encoding, where each of the distinct categories are assigned an arbitrary integer label and each value is transformed into the corresponding integer label. This approach leads to problematic performance because the magnitude of the value doesn't have any real meaning in the sense that a bigger number is not necessarily better. Another approach is one-hot encoding where each value is transformed into a vector whose dimensionality is the number of unique categories and the coordinate corresponding to the category membership of the value is set as 1, and all other entries are 0.

The approach we used is called mean encoding. Mean encoding works feature-wise in the following fashion,

1. Reduce the number of unique categories for each feature to 10, where all data points that didn't belong to any of the 10 categories were set as 'oth'
2. Each value is ordinal encoded
3. Each column is grouped by the ordinal encoded value
4. The average of the class label (1 or 0) is calculated for the group
5. A dictionary is created with the raw category as the key, and the label mean as the value

This is procedure is performed on the training set to create the mean encoding dictionary. Step 1 is performed on the testing set using the unique categories from the training set. Subsequently, the testing set is transformed using the mean encoding dictionary built from the training set. The results of mean encoding are that each categorical value is mapped to a number between 0 and 1 representing the probability that a data point with that value has a label value of 1. Mean encoding is a meaningful way encode categorical data, and corresponds to superior serparability of classes relative to ordinal encoding.

4.3.4 Machine Learning Pipeline

Four *candidate* machine learning models were initially selected, logistic regression, random forest classifier, eXtreme Gradient Boosting (XGBoost) classifier, and an ensemble of the three. These candidate models were chosen based on prior unpublished work with batch 1. 5-fold cross validation was repeatedly performed on the training set. This step allowed us to choose optimal parameters and select a model. From the results of cross validation, a stand-alone XGBoost model was chosen with a learning rate of 1×10^{-3} and 5000 esitmators. This resulted in a 5% accuracy increase relative to the base XGBoost model on cross validation.

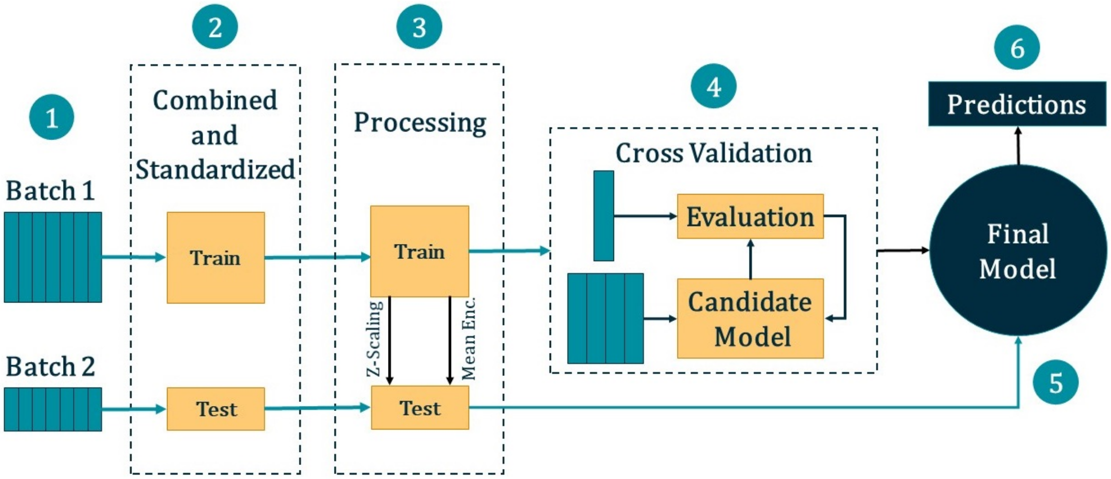


Figure 4.1. The figure shows the machine learning/data pipeline. **1)** 2 batches of data are obtained. **2)** Individual records are combined and variables/variable names are made uniform between the batches. **3)** Features are processed for admission to the ML model. Information only flows from train to test. **4)** Cross validation is performed. A model and optimal parameters are chosen. **5)** The final trained model emerges. **6)** The final model predicts the class membership of testing data.

The features used in the final model were chosen based on prior work and exploratory data analysis using the training set (not the testing set). Specifically, a mix of pearson correlation between each feature and the target, and performance on cross validation was used to select the final features.

Table 4.2. The final set of features used in the model are shown

Feature	Meaning	Type	Processing
hsGPA	High school GPA	Numerical	Z-scaled
hsMathGPA	High school math GPA	Numerical	Z-scaled
hsPhysGPA	High school physics GPA	Numerical	Z-scaled
ACT/SAT Math	ACT/SAT math score	Numerical	Z-scaled
ACT/SAT Non-Math	ACT/SAT non-math score	Numerical	Z-scaled
AP Math Score	Average AP math score	Numerical	Z-scaled
collegeGPA	most recent prior college GPA	Numerical	Z-scaled
AP Math	Number of years of AP math taken	Categorical	Mean encoded
hsMathYr	Number of years of HS math taken	Categorical	Mean encoded
AP Math	Number of years of AP math taken	Categorical	Mean encoded
AP Phys	Number of year of AP physics taken	Categorical	Mean encoded
gender	Student's gender	Categorical	Mean encoded
ethnicity	Student's ethnicity	Categorical	Mean encoded
repeatIND	If student is retaking the course	Categorical	Mean encoded
studentClassification	Student's year in college	Categorical	Mean encoded
firstGenCollege	First gen. college status	Categorical	Mean encoded

The model was trained using using the training set with variables in table 4.2, and the model was used to predict the class membership of the testing data.

4.4 Results

In this section, we will present the results of the classification task and provide a thorough discussion of the classification error. We begin by defining some useful metrics. We partitioned the group in two classes. The class designated as class 1 were the students that scored 'C+' or lower as their final grade in the course. This group would be the high risk group. Likewise for students who score 'B-' or better, class label 0, and low risk. So, we can make the following definitions:

This table (Table 4.3) is also called a *confusion matrix*, which we will be using later in this section.

Table 4.3. Table defines true positve (t_p), true negative (t_n), false positive (f_p), and false negative (f_n) classifications

	Predicted High Risk	Predicted Low Risk
Actual High Risk	t_p	f_n
Actual Low Risk	f_p	t_n

In terms of the definitions (Table 4.3) we can define several other useful quantities. *Precision* (P) is calculated for each class and is a measure of the accuracy of the predictions made by the model. In other words, what fraction of the classifications of low-risk were correct? What fraction of classifications of high-risk made by the model were correct?

$$P_1 = \frac{t_p}{t_p + f_p} \qquad P_0 = \frac{t_n}{t_n + f_n} \tag{4.2}$$

Recall (R) is calculated for each class and indicates what fraction of each class were correctly predicted. In other words, what fraction of high-risk students were detected by the model? What fraction of low-risk students were detected by the model?

$$R_1 = \frac{t_p}{t_p + f_n} \qquad R_0 = \frac{t_n}{t_n + f_p} \tag{4.3}$$

Finally, the overall accuracy of the classifier is the number of correct predictions, divided by the total number of classifications

$$Acc = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{4.4}$$

The classification trial was repeated three times. The three trials were independent in the sense that each time the model was trained on the data, predictions were made. However, the classification and the associated metrics were identical among the three trials. The results of the classification were as follows,

Table 4.4. Table shows the results of the experiment. Class membership, precision, recall, number of specimens (N), and Overall Accuracy (Accuracy)

Class	Precision	Recall	N	Accuracy
0	0.78	0.74	372	——
1	0.70	0.76	304	——
——	——	——	——	0.75

Table 4.5. Table shows the confusion matrix for this experiment

	Predicted High Risk	Predicted Low Risk	N
Actual High Risk	**231**	**73**	304
Actual Low Risk	**97**	**275**	372
Totals	328	348	676

The normalized error is calculated separately for each grade level as:

$$\overline{E}_{grade} = \frac{N_{error}}{N_{total}} \tag{4.5}$$

Where N_{error} is the number of miscategorized samples, and N_{total} is the total number of samples from that grade level. For example, if there are 75 total students that earned a grade of 'C+', and 30 of them were miscategorized (categorized as low-risk), then the normalized error for the grade 'C+' would be

$$\overline{E}_{C+} = \frac{30}{75} = 0.4 \tag{4.6}$$

We also define the accuracy by grade level as,

$$A_{grade} = 1 - \overline{E}_{grade} \tag{4.7}$$

Note: This metric is technically *recall*, but we re-framed it as accuracy to avoid confusion with the above definitions. Also, since it is defined in terms of classification error, the name

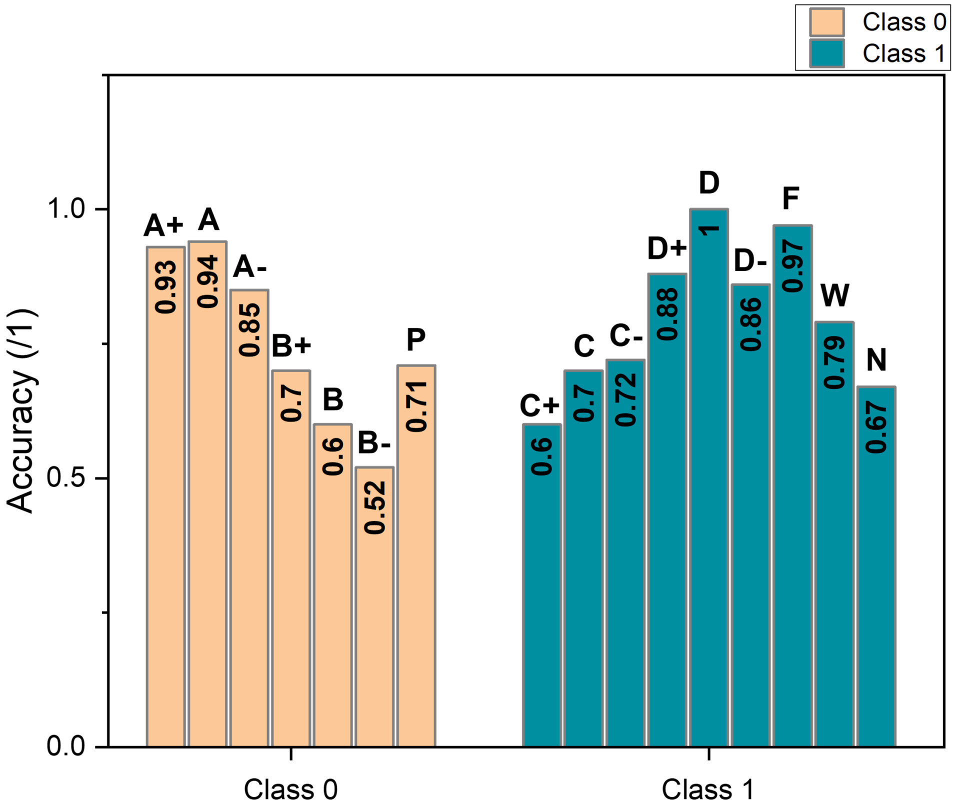


Figure 4.2. The accuracy by class for each grade level is shown.

accuracy is sensible. Finally, another way we can examine to facilitate later discussion is the prediction accuracy of the 'A' students, the 'BC' students, and the 'DFW' students.

Table 4.6. Table shows the accuracy of A, BC, and DFW students

Grade Category	Accuracy
A	0.92
BC	0.67
DFW	0.88

4.5 Discussion

In this section we will provide a thorough examination of results through the lens of the research questions **RQ1**, **RQ2**, and **RQ3**. We begin by addressing **RQ1**, which is how we should define the high and low risk groups to minimize *false negative errors*.

A fundamental part of the machine learning pipeline is to formulate a problem that is solvable, interpretable, and has useful results. It is well known that imbalanced class numbers significantly reduce the performance of machine learning models. Class imbalance causes a failure of the algorithm to *learn* the underlying distributions in the data [79], and thus produces poor results in many cases.

The results of Yang et. al. [2] were quite poor when using only instutional variables. This was due to the large class imbalance in the way that they formulated their *learning* problem. However, it should be noted that performance drastically improved by the addition of several in-class variables later in the experiment.

We believe by defining the classes as in [1], that we derive a very solvable problem whose results are still useful. Failure of the course isn't the only adverse outcome related to a "low" grade. Many of the students taking introductory physics are doing so for admission into other programs. A significant portion of this population will be applying to medical or pharmacy programs, which are very competitive. For this reason, a grade of 'C+' or below could present a significant obstacle to the students' later plans. Additionally, formulating the classes in this way allows us to detect the most at-risk students in a more timely fashion before in-class variables are available to be used in the model.

RQ2 is concerned with the ability of a model trained with this data to accurately predict at-risk students. Our classification accuracy of 75% is not especially compelling. However, as table 4.2 and table 4.6 show, the prediction accuracy does not tell the whole story. The overall accuracy is reduced because the model struggles with students near the class margin. In other words, students who got a B could have certainly gotten a C instead, and likewise students who earned a C could have instead earned a B.

It is notable that the model does very well at detecting the most at-risk students, the ones in the 'DFW' category. If intervention can be implemented in a way that is not excessively burdensome to the students, then instructors might find this error acceptable in the very beginning of the course until additional data can be added to the model to improve the precision as in [1], [2]. For the price of miscategorizing students on the margins, instructors are able to get a good indication of students that are at serious risk of adverse outcomes.

In order to gain insight into why the model struggles with students on the margin we used L.I.M.E. (Local Interpretable Model-agnostic Explanations). L.I.M.E. is a package in Python and allows the experimenter to look at individual classifications made by the model and examine the reason behind the model's prediction. Obviously since the L.I.M.E. package works on indivdual predictions, and there are 676 predictions, we cannot examine them all. However, even looking at a subset of the predictions can be informative. Students were generally categorized according to academic features. This was true regardless of grade level ('A','B',...,'F'). The most important features selected by the model were college gpa, ACT/SAT math score, HS GPA, and AP math score. This finding is in agreement with [2]. Students that were categorized as low risk had scores well above the mean for these features, and students that were correctly categorized as high risk had scores that were below the mean. The model struggled on students that were marginal and had a number of conflicting indicators. For example, if the student had poor HS attributes, but has a high college GPA. Categorical features such as ethnicity and gender were much less important but non-negligible. It seems that the model uses these *auxillary* features as "tie-breakers" in borderline cases.

RQ3 deals with assessing the model in the tripartite framework [48]. The technical feature is a subjective measure that rates the accessibility of technology to students and teachers. Due to the amount of data processing that was required for this project, our model would rate low to moderate. If an instructor is competent in machine learning/data engineering it is conceivable that they could implement a model of this type in their course. This technology could be made much more accessible if data analysts could work with instructors to make the data more uniform in terms of the variables included and the variable names. Also, analysts may be able to assist with some of the processing tasks. If this aid is available, then this method of predicting student risk could rate moderate to high.

The validity feature deals with how the model is validated. We used k-fold cross validation which is an industry best practice, and the results of cross validation were consistent with the results of the final model. However, it is difficult to make strong claims about the validity of the model until it is tested on more *unseen* data from the same poplulation or different populations (generalization). For that reason, our work would rate moderately on that metric.

The final piece is the pedagogical feature. Our work rates moderate to high here. We have shown that our model has very strong predictive power for the highest risk category ('DFW'). If the model was used properly, it would allow for very early intervention for the most at-risk students.

Another feature not mentioned in [48] is the ethical feature. Our work rates moderate to high in this category. Arguably the model would rate high here because it focuses on academic features, and does not use demographic features (primarily) to classify students as high risk. The model would be unethical if it relied on features such as gender or race to make it predictions. Use of this model in a real classroom would be slightly unethical in the sense that it miscategorizes one out of every three students in the 'BC' group.

Much of the harm associated with miscategorizing the 'BC' student could be mitigated by thoughtful implementation of this model. Students should volunteer for inclusion in any prediction about their future outcome in the course. Furthermore, the instructor should openly acknowledge the limitations of the model in discussions with students. For example, telling students that were categorized as high-risk by the model, that there is 30% error rate associated with predictions of high risk. The instructor should also mention that model indentifies 'DFW' students with very high accuracy. As mentioned previously, prediction power can rapidly increase over time. It might also bolster confidence for students who were previously categorized as high risk to change their risk category through their own work in the course.

4.6 Conclusion

In this work we obtained data from the registrar which included information about a student's academic past, and their demographics. This data was used to train a machine learning algorithm. The trained algorithm was used to predict the students that are at risk of adverse outcomes. The model has reasonable, but not perfect, performance on the dataset as a whole. In the results and discussion sections we provided a thorough discussion of the strengths and the weaknesses of the model.

The model has high performance at detecting the most at-risk students. While the model doesn't directly identify the student as likely to earn a 'DFW' score, it correctly identifies them as 'at-risk'. The model also miscategorizes a significant number of high risk students as low risk, and vice-versa. These miscategorized students are mostly on the margins. This doesn't mean that it is acceptable that they are not correctly categorized. It only means that instructors should be aware of this potential problem and have mitigation strategies in mind until more data can be added to the model to improve prediction.

We also showed that the model could be used ethically, since it doesn't discriminate against students by using features such as gender or ethnicity to make its predictions. We also argued that prior work [1], [2] showed that prediction power can be greatly improved by the addition of in-class variables gathered in the first weeks of class. This allows for predictions that are both timely and accurate. Furthermore, this model could be improved by the addition of other features such as the student scores on a diagnostic instrument such as the Force Concept Inventory [70].

5. USING NATURAL LANGUAGE PROCESSING TO PREDICT STUDENT PROBLEM SOLVING PERFORMANCE

5.1 ABSTRACT

In this work we report on a pilot study where we used machine learning to predict whether students will correctly solve the classic “ballistic pendulum” problem based on an essay written by students elucidating their approach to solving the problem. Specifically, students were asked to describe the “principles, assumptions, and approximations” they used to solve the problem. Student essays were codified using the practices of natural language processing. Essays from two non-consecutive semesters were used for training/validation (N = 1441) and testing (N=1480). The final model used to make predictions was an ensemble classification scheme using random forest, eXtreme Gradient Boosting classifier (XGBoost), and logistic regression as estimators. Our accuracy in predicting students’ correctness was around 80% with slightly higher accuracy in identifying students who incorrectly solved the problem and slightly lower in predicting student who correctly solved the problem.

5.2 Introduction

Research has shown that facilitating students to attend to the underlying concepts and principles needed to solve a problem improve problem solving performance [80], [81]. We implemented strategy writing [81] in a pilot study with students in a calculus-based physics course at a large public mid-western university. Students were asked to write an essay describing their strategy for solving a problem. Their essays were analyzed using Natural Language Processing (NLP) to determine whether they could predict the ground truth label i.e. the correctness of the student's answer to the problem.

NLP is a branch of artificial intelligence (AI) in which computers perform operations on human language. NLP has numerous applications such as determining the sentiment of tweets; chatbots/assistants which perform speech recognition/generation; and machine text translation. Classification in NLP is at the intersection of machine learning and NLP. Machine learning (ML) can be thought of as a collection of methods where a statistical model is developed that maps numerical data on to a target variable (label). A ML algorithm is trained when an objective function which quantifies the error made by incorrect predictions is minimized with respect to the model's parameters (e.g. weights and biases in the case of multiple linear regression). The trained model is then used to predict the class membership of unseen data known as a testing set. The fundamental rule of ML is testing data is not used for training or any manner of model parameter tuning.

In this work we report on the use of NLP to predict whether students in a first semester calculus-based course would correctly solve a problem (Fig. 1) during a quiz taken in lab. We asked students to write an essay describing their strategy for solving the problem, including underlying principles used, and objects in the system/surroundings. Data were labeled 0/1 based on whether students solved the problem incorrectly/correctly. This work was exploratory in nature to determine how well we could make accurate predictions. Our vision for the future of this work is a platform to provide in-situ feedback to improve student learning.

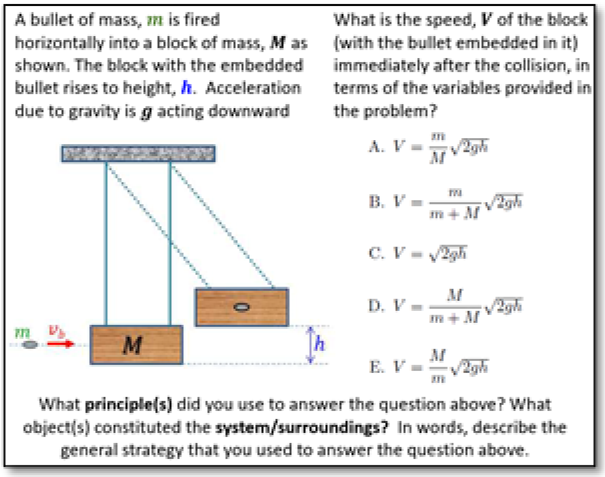
The text data from the essay were transformed using the term frequency-inverse document frequency (TFIDF) method. We constructed a ML model using the Scikitlearn [82] library in Python. The final prediction model was a hard voting scheme using Random

Forest [83], Logistic Regression, and eXtreme Gradient Boosting classifier [84] as estimators. We used data from Spring 2020 for model training and general validation, and data from Spring 2021 for testing. More details are presented in the following sections. We addressed the following question.

1. **Research Question**: With what accuracy can we predict if a student will correctly

5.3 Methods

Students completed the task shown below (fig. 5.1) on Quiz 3, which was administered in Week 7 of the semester. The quiz was administered in a sterile environment where notes and collaboration were not allowed. We chose this problem because it is a well-known problem in introductory physics that students have difficulties with.



A bullet of mass, m is fired horizontally into a block of mass, M as shown. The block with the embedded bullet rises to height, h. Acceleration due to gravity is g acting downward

What is the speed, V of the block (with the bullet embedded in it) immediately after the collision, in terms of the variables provided in the problem?

A. $V = \frac{m}{M}\sqrt{2gh}$

B. $V = \frac{m}{m+M}\sqrt{2gh}$

C. $V = \sqrt{2gh}$

D. $V = \frac{M}{m+M}\sqrt{2gh}$

E. $V = \frac{M}{m}\sqrt{2gh}$

What **principle(s)** did you use to answer the question above? What object(s) constituted the **system/surroundings?** In words, describe the general strategy that you used to answer the question above.

Figure 5.1. Problem solved by students in online Quiz 3 in Week 7

5.3.1 Data

The descriptive statistics for the word length of the essay data are shown in Table I. A thorough analysis of the differences between the words and phrases used by each group is beyond the scope of this paper. There is no significant difference in essay length between the correct and incorrect responses, or between the data sets.

Table 5.1. Table shows descriptive statistics for the essays

Data Set	Mean ± S.D.	Median
Spring 2020	correct ($N_1 = 703$) 57.9 ± 31.2	51
(training)	incorrect($N_0 = 738$) 56.5 ± 29.9	51
Spring 2021	correct ($N_1 = 679$) 60.2 ± 32.0	55
(testing)	incorrect ($N_0 = 801$) 59.8 ± 37.3	52

5.3.2 Text Processing

Text Cleaning

The essays from both sets were cleaned using a function in Python, that removes unimportant commonly used words (stop words) [85] to reduce noise, as well as punctuation, numbers, and equations which some students (6.1% in training, 4.5% in testing set) included in the essay. Finally, the essays were spell checked using a context-unaware spell checker from the textblob [86] library.

TFIDF Transformation

ML algorithms cannot perform computation on raw text. Most standard methods in NLP involve transforming text into a vector. The simplest approach is the bag-of-words model in which text is transformed into a vector of dimensionality equal to the number of unique words in the corpa and whose components are the word counts in a particular corpus. A higher level of sophistication is the TF-IDF transformation, which converts each essay (corpus) into a vector whose dimensionality is the number of unique words in all the essays (corpa). The components of each vector are a calculated score for each unique word in the corpa based on its frequency of appearance in that corpus and inverse frequency in the corpa:

$$W(t, d, D) = f_{t,d} log \left(\frac{N}{n_t} \right) \tag{5.1}$$

The TFIDF score, W, for each word, t, is calculated corpus-wise for each document d in the corpa D. W is large for words with a high frequency (f) appearing in a small number of documents (n_t). W is low for words that have low frequency appearing in a large number of documents.

5.3.3 Prediction Model

The prediction model uses three independent estimators, Random Forest [83], eXtreme Gradient Boosting (XGBoost) [84], and Logistic Regression. The predictions emerging from these algorithms are combined to make a single final prediction, a scheme known as ensemble learning.

Random Forest Classifier

A decision tree is a flowchart like structure where a datum is classified after passing through a network of nodes representing features of the model. In some cases, the decision tree can be conceptualized as a series of yes/no questions that ultimately results in a classification [87]. Decision trees are robust to irrelevant features (noise) and are capable of learning complex patterns. However, they tend to learn the training set very well while struggling with unseen testing data (overfitting). The random forest classifier is an ensemble (forest) of decision trees [83]. Each tree in the forest is built by randomly sampling the training data with replacement, a method known as bootstrap aggregation, and using a random subset of the features (variables) to make predictions. The final classification is the majority vote of all the trees. This has the effect of reducing overfitting relative to a single decision tree by producing a series of weak uncorrelated learners which averaged together make more accurate predictions [88].

eXtreme Gradient Boosting Classifier (XGBoost)

Boosting is a technique whereby the classifier learns from its mistakes (incorrect predictions) [84]. The version of XGBoost used in this work is based on the random forest classifier. XGBoost uses boosted tree learning to improve upon the consistently high performance of random forest. The goal of XGBoost is to learn a decision function (classifier) that encapsulates the structure and function of a random forest. Boosting happens in iterations called boosting rounds. The decision function is initialized to a constant value, obtained by solving an optimization problem. During each of the m subsequent boosting rounds the decision function is updated recursively to correct mistakes made in the previous round. This scheme results in a classification algorithm that is robust to overfitting but can be susceptible to outliers [89]. For labeled data $\{x_{\mathrm{i}}, y_{\mathrm{i}}\}$, the decision function F_m after the m-th boosting round, and the regularization term h_m:

$$F_m\left(x\right) = F_{m-1}\left(x\right) + \alpha_m h_m(x, r_{m-1}) \tag{5.2}$$

While the usual gradient descent algorithm that is at the heart of machine learning aims to minimize the objective function with respect to the parameters of the decision function, gradient boosting endeavors to minimize the objective function with respect to the decision function.

Logistic Regression

In logistic regression we predict samples using the sigmoid function:

$$h\left(x\right) = \frac{1}{1 + \mathrm{e}^{-\theta x}} \tag{5.3}$$

Where θ is a vector of weights and biases (high dimensional analog to slope and intercept) and x is a feature vector (data point). The vector θ is obtained by minimizing the log-loss objective function with respect to θ.

$$J\left(\theta\right) = -\frac{1}{m}\sum_{\mathrm{i}=1}^{m}\left[y^{(\mathrm{i})} \times \log\left(h_\theta\left(x^{(\mathrm{i})}\right)\right) + \left(1 - y^{(\mathrm{i})}\right) \times \log\left(1 - h_\theta\left(x^{(\mathrm{i})}\right)\right)\right] \tag{5.4}$$

The sigmoid function is a continuous valued function bounded on (0,1). When making a binary classification, a thresholded decision function $h'(x)$ is used such that:

$$h'\left(x\right) = \begin{cases} 1 & \text{if } \; h(x) \geq 0.50 \\ 0 & \text{else} \end{cases} \tag{5.5}$$

5.3.4 Training, Testing, and Validation

Model training is the process (Fig. 3) of using the training data to select the optimal parameters for a given model. The optimal parameters are usually determined by minimizing an objective function with respect to the model parameters. This gives a candidate model. The success of a model is determined by its ability to correctly classify unseen data. The hypothetical scenario is that the testing set is not available to you when you create the model, and it will be used in production to classify new data in real time. Thus, it is necessary to validate the model before production on some data that was not used during training (validation set).

In k-fold validation, we split all of the data into k equal sized partitions. k-1 sets are used for training and the remaining set is used for testing. This is repeated until all k sets have been used in training and testing. The accuracy is averaged across the k trials.

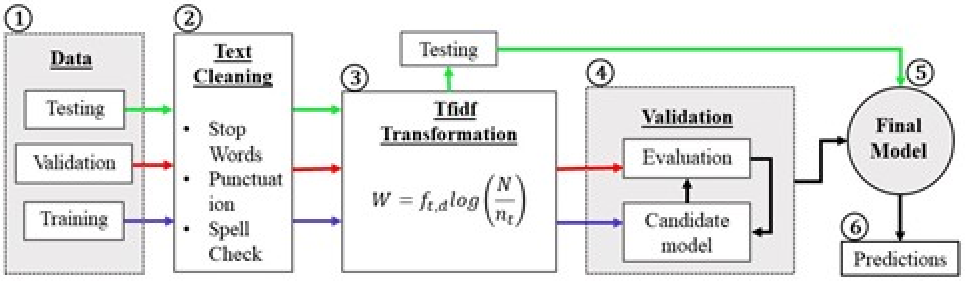


Figure 5.2. A diagram showing the machine learning workflow. (1) Training data (blue), validation data (red), and testing data (green) are processed by (2) removing stop words, punctuation, and checking spelling. (3) A Tfidf transformer object is fitted to the training data and used to transform training, validation, testing sets. The testing set is put aside. The training set is used to train a candidate model, and the candidate model is evaluated on the validation set. (4) The model is tuned in a feedback loop to improve classification performance on the validation set. The process continues until performance is saturated and the final model (5) emerges. The training and validation sets are used to train the final model and (6) predictions are made on the testing set

5.4 Results

The classification accuracy is an important metric by which to judge the performance of the prediction model. However, accuracy should not be considered in isolation. Other important metrics to consider are precision, recall, and F-score.

We define a true positive (t_p) classification as a student who is labeled '1' and is predicted as '1', likewise a false positive (f_p) classification is a student is labeled as '0' but predicted as '1'. We define a true negative (t_n) as a student who is labeled as '0' and predicted as '0', likewise a false negative (f_n) is a student who is labeled '1' but predicted as '0'.

Precision is the fraction of correct classifications made by the classifier.

$$P_1 = \frac{t_p}{t_p + f_p} \qquad P_0 = \frac{t_n}{t_n + f_n} \tag{5.6}$$

Recall is the fraction of each population correctly identified by the classifier.

$$R_1 = \frac{t_p}{t_p + f_n} \qquad R_0 = \frac{t_n}{t_n + f_p} \tag{5.7}$$

The F-score is the harmonic mean of precision and recall. F-score is a balanced metric to determine the overall quality of the classifier.

$$F_1 = 2\frac{P_1 R_1}{P_1 + R_1} \qquad F_0 = 2\frac{P_0 R_0}{P_0 + R_0} \tag{5.8}$$

Cohen's kappa [90] is a measure of agreement between raters, controlling for agreement by chance.

$$\kappa = \frac{p_0 - p_\mathrm{e}}{1 - p_\mathrm{e}} \tag{5.9}$$

Where p_0 is the observed agreement between raters, and p_e is the probability of agreement by chance. The results of the classification are in Tables 5.2 and 5.3 below. Finally, Cohen's kappa was calculated to be $\kappa = 0.594$.

Table 5.2. Precision, Recall, and F-score

Class	N	Precision	Recall	F-Score
0	801	0.79	0.87	0.82
1	679	0.82	0.77	0.77

Table 5.3. Confusion matrix: Correct predictions are on the diagonal. Incorrect predictions are off diagonal.

	Predicted Negative	Predicted Positive	Accuracy
Actual Negative	695	106	—
Actual Positive	190	489	—
—	—	—	0.8

5.5 Discussion

A proposed instrument for essay scoring should only be deployed if it is shown to be valid, fair, and reliable. A method is considered valid if it measures what it claims to measure. A method is fair if it does not unfairly penalize correct responses, and it is reliable if the results are repeatable [91]. It is difficult to gauge the validity of this model without a direct comparison with other models which perform the same function. Many projects that attempt automatic essay scoring (AES) use comparison with human raters as a metric [92].

A competition among commercial AES vendors used eight student essay corpa from six member states of the Race-To-The-Top assessment consortium as a dataset [92]. Students wrote persuasive, expository, narrative, and source-based essays (where they formulated an argument based on a passage). This dataset used the state adjudicated score conferred by human scorers (resolved score) as the ground truth (label) and compared the performance between different proprietary scoring engines. A metric used in this study is percent agreement between computer scoring systems and the resolved score. Percent agreement (identical to accuracy) is the percentage of times the resolved score and the computer score were identical. The percent agreement of the scoring engines ranged from 0.29 to 0.76, and the Cohen's κ ranged from 0.04 to 0.84 across eight datasets. Thus, our results (accuracy = 0.80, and Cohen's $\kappa = 0.594$) are within the range of proprietary scoring engines used in [92].

A key difference with our study is that in [92] the essays themselves were scored by a multi-point rubric, while we did not score the essays per se, rather we used problem correctness (0/1) as a proxy for scoring of the essays themselves. It is also worth noting that the scoring engines in [92] had high performance on "adjacent agreement" when the computer score was within 2 points of the resolved score on a rubric of 8 points (maximum). There is no way to directly compare our results on this metric due to the differences in essay scoring.

Presently, there is not enough information to establish that our prediction model is valid for scoring student essays themselves. However, the goal of the present study was to use the strategy essay to predict if the student will correctly solve a problem. If we could substantially reduce the error rate, this model could be useful to provide feedback to students so they can correct errors before submission.

In regards to fairness, about 20% of students were incorrectly scored, out of which 13% were predicted incorrect despite solving the problem correctly. Finally, since we currently only have two sets of data to work with, we cannot establish the reliability of this model.

5.6 Conclusions, Limitations & Implications

Despite the shortcomings of our classification scheme these results are promising since the model is able to predict, based on the strategy essay written by a student, whether or not the student has answered the problem correctly with 80% accuracy. For the purposes of predicting incorrect answers, the prediction rate is 87%.

This study has the following limitations. First these results leave room for improvement in accuracy and fairness, which could be achieved with a larger training set and more powerful state-of-the-art machine learning methods, such as deep learning. Second, the study used only a single problem that required students to determine their answer in symbolic representation using a multiple-choice format. Therefore, the results are not generalizable to problems in other formats and representations, not to mention other topical areas in introductory physics. Finally, the study did not score the essays themselves, rather it predicted the scores of the problem that students wrote the strategy essay for, and they may have written this essay not necessarily before solving the problem. Future studies will use human raters to score the essays based on the validity of the outlined approach, and therefore the likelihood that the strategy will lead to a correct solution.

Despite these limitations, the study has several implications for research and education. This study provides proof-of-concept that it is possible to predict students' correctness of a problem with a high degree of accuracy, based on the essay they have written describing their strategy to solve the problem. Research has shown that asking students to describe their strategies for solving problems can be useful in helping them develop more expert-like problem solving strategies [80].

However, past studies did not provide feedback to students on their strategy writing. The time cost of providing such feedback, especially in large enrollment introductory classes can be prohibitive. The results of this study are promising because they provide proof of concept that it might be possible, using NLP methods to provide students feedback on their strategy writing in real time, thereby giving them the opportunity to reflect on, and if necessary, alter their problem-solving strategy before they apply it to solve the problem. Such a system would also allow us to investigate whether real time strategy feedback can improve students' metacognitive skills and make them more expert-like problem solvers in the future.

6. CREATION OF AN ADAPTIVE ONLINE INSTRUCTIONAL MODULE

6.1 Abstract

In this study we developed an online instructional module to teach the concepts of force and motion. The module was created in two versions meant to optimize instruction for students based on their level of physics domain knowledge according to the expertise reversal effect [12], [93], [94]. Specifically, a high level guidance (HLG) version and a low level guidance (LLG) version. The HLG version provides guiding features, such as continuous animations and voice voice narration. The low level guidance version is self-paced and shows pop-up content as the user interacts with the page.

The module was deployed to N=378 students from a first year algebra based mechanics course. The module was hosted on a web server and students used the module in a location and time of their choosing. The number of participants was reduced to N=171 when filtering students who used the modules in a way that indicated they were not putting forth ”honest effort” based on timing metrics. The results indicate that both versions of the module were generally effective in teaching students the principles of force and motion. However, we failed to show that the modules provided adaptive instruction to students based on their domain knowledge level.

6.2 Introduction

Online learning has been a common component of many classrooms for nearly 20 years. Meanwhile, institutions have been slow to use these systems to their full potential [95]. The global pandemic has caused institutions to re-examine how online learning should be used to deliver course content. Additionally, a larger number of students are choosing online education. This cohort finds the traditional classroom ”restrictive, inflexible, and impractical” [96]. Alternatively, the online classroom allows students (some non-traditional) with a tight schedule access to higher education[96].

E-learning systems should be engaging, appropriately paced, and provide timely and useful feedback[97], [98]. Moreover, the same instructional considerations that are employed when designing in-person course content should be addressed when designing online educational materials. Guiding principles for the design of instructional materials can be derived from cognitive load theory [3]–[5].

Cognitive load refers to the utilization of working memory rescources during learning and problem solving. Intinsic cognitive load (ICL) is presented by the inherent complexity of the material. ICL can differ for given material based on the characteristics of individual learners. Specifically, material may present different levels of ICL to experts and novices. Extraneous cognitive load (ECL) is cognitive load that arises from utilizing cognitve resources on activites that are not productive for learning and problem solving. ECL can result from sub-optimal instructional design. For example forcing learners to mentally integrate multiple streams of information produces a high ECL environment relative to providing the same information in a single source [17]. Germane cognitive load (GCL) refers to the working memory resources used to build schema. GCL is incurred by the learner as they use cognitive resources to process the intrinsic load. ECL reduces the efficiency of the ICL processing. Mayer’s principles of multimedia learning [99] distills the insights of cognitive load theory and related work, and provides a clear set of principles for instructors.

According to Mayer, multimedia instructional materials are words and pictures used to convey an understanding of a concept or several concepts. Words can be written or spoken, and pictures can be still images, diagrams, graphs, videos, or animations. These linguistic

Table 6.1. Mayer's Principles of Multimedia Learning as written in [100]

Principle	Description
Coherence Principle	”People learn better when extraneous words, pictures and sounds are excluded rather than included.”
Signaling Principle	”People learn better when cues that highlight the organization of the essential material are added”
Redundancy Principle	”People learn better from graphics and narration than from graphics, narration and on-screen text.”
Spatial Contiguity Principle	”People learn better when corresponding words and pictures are presented near each other rather than far on the screen.”
Temporal Contiguity Principle	”People learn better when corresponding words and pictures are presented simultaneously rather than successively.”
Segmenting Principle	”People learn better from a multimedia lesson that is presented in user-paced segments rather than as a continuous unit.
Pre-training Principle	”People learn better from a multimedia lesson when they know the names and characteristics of the main concepts.”
Modality Principle	”People learn better when words are presented as narration rather than as on-screen text.”
Multimedia Principle	”People learn better from words and pictures than from words alone.”
Personalization Principle	”People learn better from multimedia lessons when words are in conversational style rather than formal style.”
Voice Principle	”People learn better when the narration in multimedia lessons is spoken in a friendly human voice rather than a machine voice.”
Image Principle	”People do not necessarily learn better from a multimedia lesson when the speaker's image is added to the screen.”

and visual components are used in such a way that when the student receives instruction in this modality, they should build a mental model. From this perspective, the primary goal of multimedia instruction is schema development. Therefore, multimedia learning has taken place when the learner has developed a mental model as a result of multimedia lessons [99].

This work seeks to build an online instructional module to teach the principles of force and motion to students from an alegbra based introductory physics course at a large mid-western university. We made a web application (webapp) that delivered content in two versions. One version designated as the high-level guidance (HLG) version used an AI generated voice to deliver narration, which was synched to animation to produce the effect of a video that rendered in the participant's web browser. Another version designated as the low level guidance (LLG) version had on screen text and still pictures appear when participants hovered their mouse over certain content areas that were prominent on the screen. Content displayed in this manner will hereafter be referred to as pop-up content.

The expertise reversal effect (ERE) [12], [93], [94] examines the differences between learners based on their level of expertise in a given domain (domain knowledge). The HLG version of the module was intended to benefit learners with low prior knowledge (LPK). Likewise, the LLG version is intended for learners with a relatively high level of prior knowledge (HPK). The principal difference between HPK and LPK learners is the existence of knowledge structures , called *schema*, in their long term memory that are vitally important in learning and problem solving. Schema are generally built incrementally as the learner gains new experience. Thereafter, new knowledge has to be reconciled with and integrated into existing schema.

Schema that are responsible for directing attention to relevant pieces of information, disregarding irrelevant information, and constructing a solution to the problem at hand are missing in the LPK learner. The LPK learner, lacking relevant schema, resorts to ”mostly random and cognitively inefficient” activities[94]. Furthermore, ”direct and explicit” instruction can be a substitute for these knowledge structures in learning and problem solving activities[94]. In otherwords, LPK students can compensate for their lack of schema (to an extent) by being provided explicit instructions.

The guiding features present in the HLG version designed to minimize CL for LPK learners, must also be processed in the working memory of HPK learners. However, already posessing the necessary knowledge (as schemas), this instructional guidance is not only unnecessary but harmful[12], [94], [101]. In other words, instructional guidance presents an ECL to HPK learners if present, and an ECL to LPK learners if absent [94].

According to cognitive load theory and related work, the LLG version of this module should present a working memory overload for LPK learners. Replacing the majority of on-screen text with auditory narration in the HLG version allows the involvement of auditory channels of working memory [14], [102] reducing the load on visual channels. Meanwhile, HPK learners are able to leverage pre-existing mental models to mitigate cognitive load imposed by heavy loading on visual channels.

In the LLG version, learners process the text in the bulleted list and the ”pop-up” content displayed when they hover their mouse over the item serially, rather than simultaneously, which eliminates the need to mentally integrate the separate streams of information as in the split-attention effect. This approach has been found to improve learning outcomes relative to a split-soure format or an integrated format [103]. Additionally, the use of pop-up elements in the LLG version leads to educational content that is transient in nature. Transient information can itself be a source of cognitive load. Transient information presents a disproportionately large ECL for the LPK learner relative to the HPK learner. Both learners must hold this information while processing the lesson. The HPK learner experiences a smaller ECL in general due schema guiding their processing of the lesson and any additional ECL from transient material is nominal. Meanwhile the LPK learner processes the lesson inefficiently and feels a greater impact from transient items [104].

The HLG version reduces cognitive load by replacing much of the on-screen text with narration as in the modality effect (tab. 6.1). The HLG version further reduced ECL for LPK learners by avoiding transient materials, opting instead for continuous animations [104]. Both versions of the module are structured in sections and subsections that the user is allowed to navigate at their own pace, respecting the segmenting principle (tab. 6.1). Deploying this module after the concepts of force and motion are covered in their physics course complies with the pre-training principle (tab. 6.1). Both versions of the module were designed to reduce extraneous contents pursuant to the coherence principle (tab. 6.1).

The research questions that we addressed in this study were the following:

1. **Research Question 1 (RQ1)**: To what extent can students learn the principles of force and motion from online instructional modules?
2. **Research Question 2 (RQ2)**: How can we optimize these modules for all students using Mayer’s principles of multi-media learning?
3. **Research Question 3 (RQ3)**: How can we classify students based on domain knowledge?

4. **Research Question 4 (RQ4)**: To what extent can we further optimize these modules by providing instruction adaptively to students based on their level of domain knowledge using CLT and ERE?

In the methods and materials section we describe how the module was created. We describe the material covered and how it was formatted to teach users how to solve problems involving force and motion. We also show a side-by-side comparison between the HLG and LLG implementations of this module. In the results section, we present the results of the study. In the discussion section we evaluate our results. Finally, in the conclusion section we summarize our study and address the research questions.

6.3 Methods and Materials

In this work we developed an online instructional module (a webapp) to teach the principles of force and motion. Specifically, the webapp was intended to teach students how to solve problems on force and motion. N=378 Students voluntarily participated in the study in return for 2% of their course grade in extra credit. Students were provided a link to the module (https://www.themoduleproject.info) on the course management sytem (Brightspace). Upon logging in to the module with their school username and school ID Number, they were randomly assigned to either the HLG version or the LLG version of the module.

Both versions of the module contained four sections; a pre-test, an instrucional section, the cognitive load survey, and a post-test. Students began the module by completing a 10 question pre-test. The pre-test was the same for HLG and LLG versions of the module. Next, students were shown some directions about how to use the module. Afterward they were directed to the instructional section of the module. The same four subsections were present in both versions. Namely a subsection where the student learned vector decomposition, a subsection where the student learned to translate a problem statement into a force diagram and subsequently write an expression for the net force, another subsection where the student was instructed how to reason about the direction of acceleration (i.e. determine the y-component of the acceleration must be zero for an object in linear motion in the x-direction), and lastly an instructional subsection summarized the other three subsections and provided an example of the entire process for solving problems. The content of the four subsections was fixed between HLG and LLG, and they mainly differed by presentation of the material. Next, students from both conditions completed the judgement of learning and cognitive load survey where they answered questions designed (and validated [105]) to facilitate self-reporting of cognitive load. Finally, students took a 10 question post-test that was identical to the pre-test and identical between conditions (LLG and HLG).

The HLG version of the module was created with the LPK student in mind. The LPK student is one who has had very little prior exposure to physics and also likely has weak mathematical preparation. For this population, content was delivered as a continuous animation that was synched to an audio track. This had the effect of a video that rendered in the participant's web browser. This modality was chosen based on the theory that displaying content in this way would be more immersive than a stand-alone video being shown. The voice track was made using an AI voice that had human-like qualities.

After each of the instructional subsections were complete, a question was shown. The questions were similar to the pre-test questions and for the HLG version, process feedback was shown. Process feedback emphasizes the logical steps behind arriving at the solution. The feedback was delivered via a "video" in the style outlined in the previous paragraph.

The LLG version of the module was created for the HPK student. The HPK student is one who has had some prior exposure to physics and likely has stronger mathematical preparation than their LPK counterparts. In the LLG version, content is shown in the manner of a lecture slide with numbered bullet points. When the user hovers their mouse over each bullet point, additional content such as figures and diagrams are shown. In most cases, when the user moves their mouse off of the area, the content disappears.

After each of the instructional subsections were complete, a question was shown. The question was identical to that from the HLG condition. Upon submission of their response to the question, product feedback was shown. Product feedback just gives the correct final answer. This was shown at the bottom of the screen after submission.

More detail about the instructional section of the module and the precise differences between HLG and LLG versions will be shown in the coming subsections.

6.3.1 Vectors Subsection

The vectors subsection of the module was created to demonstrate vector decomposition, which is a necessary step for writing Newton's second law in a component form.

HLG

The HLG version provides explanations of concepts in a more basic way including more explicit instruction, than the corresponding LLG version. Compare the differences between fig. A.1 and fig. A.6 where the idea of a vector having components is explained (HLG) or stated (LLG).

The video starts and the voice provides instruction as bullet points appear on the screen. Content is delivered to the student in a fixed sequence, although they have the ability to seek forward and backward. A vector and its x and y components were presented as making a right triangle. Next, the video goes on to show that trigonometry applies to the triangle formed by a vector and its components, just as any other right triangle (fig A.2).After the vectors instructional subsection is complete, the participant goes on to answer an assessment question(fig A.3).

LLG

In the LLG version of the module, students hover their mouse over areas on the screen to reveal content (fig. A.6). When the student moves the mouse away from the item, the content disappears and the screen returns to its previous state (fig A.5).Students are next shown a still frame showing how to obtain the x and y component of a vector, given the magnitude of the vector and the angle with respect to the x axis (fig A.7). Students are presented with the same assessment question as the HLG version (fig A.3) upon completing the instruction portions of this subsection. Upon submitting their answer, they are shown product feedback (fig A.8).

6.3.2 Net Force Subsection

In this section students are instructed on how to parse a problem statement and use key information to draw a force diagram, and in-turn use the force diagram to write the net force.

HLG

The video begins by defining Newton's second law and explaining some of its properties. Specifically, that Newton's second law is a principle that tells how forces applied on an object change's the object's motion (i.e. produce acceleration). Secondly, that since Newton's second law is a vector equation, that the net force will always be in the same direction as the acceleration (fig. A.9).

Next, the definition of the net force is presented as the vector sum of all of the forces acting. Since we endeavor to write Newton's second law in a component form, we also define the x and the y component of the net force (fig A.10). Subsequently vector decomposition is revisited. We elaborate on our earlier discussion from the vectors sub-section to include a general treatment where the vector is in other quadrants of the x-y plane than quadrant 1. Also, for the cases where the angle is defined relative to the x and y axis (fig. A.11).

In the following portions of the video, we reinforce these concepts and show a concrete example of how to apply these skills. Given a problem statement, we create a vector diagram with step-by-step explanations (fig A.12) as forces are added to the diagram. With the complete force diagram, students are shown how to write the x and y components of the net force (fig. A.13). Finally, the students are shown another assessment question. They are provided process feedback on submission of their response.

LLG

The same content is covered as in the HLG version to the level that is appropriate for the LLG version. Also as in the vectors subsection the user interacts with the page by hovering their mouse over prominent areas on the screen. As in the HLG version of the net force subsection, the instruction begins by defining Newton's second law and pointing out some properties (fig A.15).

Next the net force and its components are explicitly defined (fig. A.16). After which a procedure for finding the components of forces acting in other quadrants of the x-y plane, and for angles with respect to x and y axes, is formalized (fig. A.17). In the following part of this subsection, the student is present a problem statement and a pop-down list of items. As they hover their mouse over items that are named by the forces acting in the problem, the forces appear on the set of x-y axes.

Next the student is directed to hover their mouse over the force diagram. When they hover their mouse over one of the x-y axes, the other axis disappears along with the forces acting in that other direction. Simulatenously, the net force is shown for the direction that is being interacted with (fig. A.19). Finally, the student goes on to answer an assessment question (fig. A.14). Upon submission, they are shown product feedback.

6.3.3 Acceleration Subsection

In the acceleration subsection we start with a discussion of the kinds of problems that can be solved with Newton's second law. We try to look at how forces produce acceleration, and thus change an object's motion. Finally we return to our example problem and try to reason about the direction of the acceleration. This is a necessary step because it eliminates one of the unknown quantities from an otherwise *underdetermined* system of equations.

HLG

The video begins with a brief revisitation of Newton's second law (fig A.20). A short statement on how Newton's second law can be used is given in terms of what information is needed and what information can be extracted. Next, we formally define acceleration and use Newton's second law to directly relate an object's change in velocity to the net force applied on it (fig. A.21).

Subsequently we consider some conceptual examples showing specifically how an applied net force leads to a specific final velocity (fig. A.22). We conclude this section by deducing the direction of the acceleration, and explicitly substituting this value into Newton's second law. The student then solves two assessment problems and is shown process feedback for each question (fig. A.23 and fig. A.24). The students are allowed (as with the other subsections) to revisit the instructional portions of the subsection. Students are not allowed to revisit the first question after they submit it.

LLG

This subsection starts with several list items (bullet points) that reinforce conceptual understanding of Newton's second law. One of the bullet points shows the relationship between the change in velocity and the net force (fig. A.26). In the next portion on this section, the student sees an interface with three possible initial velocities listed as well as a statement about the nature of the example and some assumptions that are made (fig. A.25). When the student hovers their mouse over one of the indicated initial velocites in the list to the left, the statement disappears and an empty set of x-y axes appears. Also, a set of buttons appear above the axes. This set of buttons blinks indicating that they are to be interacted with (fig. A.27). When they interact with these buttons additional content appears. One of the buttons shows the net force acting on the object. The next button shows the resulting acceleration. The next shows the final velocity for the object experiencing that acceleration given the initial velocity.

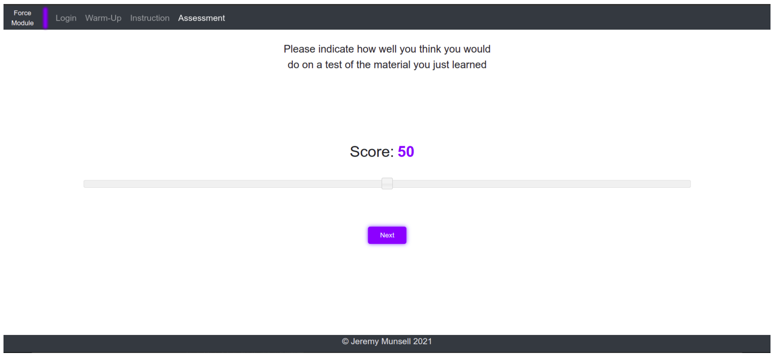
The student then moves to another portion where the direction of the acceleration is deduced. The same information coveryed by the narration in the HLG version is communicated by blocks of text in this version (fig. A.29). This subsection concludes with the student answering the same assessment questions as the HLG version (fig. A.23 and fig. A.24). Upon submitting their answer, they are shown product feedback.

6.3.4 Conclusion Subsection

In the final subsection of this module, the process of solving Newton's second is *finalized.* The example problem developed over the previous subsections is solved in a manner that is appropriate for that condition/version.

6.3.5 Judgement of Learning/Cognitive Load Survey

After completing the instructional section of the module students are shown the judgement of learning slider question (fig. 6.1) and the cognitive load survey where they self-report the cognitive load they incurred as a result of the module (tab. 6.2).



Please indicate how well you think you would
do on a test of the material you just learned

Score: 50

Figure 6.1. The judgement of learning slider question from the module

Table 6.2. Table shows the items from the cognitive load survey and their respective CL categories.

Intrinsic Load	The topic(s) covered in this activity was(were) very complex.
	The activity covered formulas that I perceived to be very complex.
	The activity covered concepts and definitions that I perceived to be very complex.
Extraneous Load	The intructions and/or explanations during the activity were very unclear.
	The instruction and/or explanations during the activity were, in terms of learning, very un-effective.
	The instruction and/or explanations during the acitivity were full of unclear language.
Germane Load	The activity really enhanced my understanding of the topic(s) covered.
	The activity really enhanced my understanding of physics.
	The activity really enhanced my understanding of the formulas covered.
	The activity really enhanced my understanding of concepts and definitions.

6.3.6 Filtering Students

The module allows students complete freedom of traversal. As a result, there is no mechanism to stop students from clicking through the module without actually viewing the material. The fastest time that a participant completed the LLG pre-test was 51 seconds, and 26 seconds in the case of the HLG pre-test. The longest time in which someone completed the pre-test in the LLG condition was 267 hours. This was very likely due to the student stopping in the middle of one of the questions, and finishing it before the due date. A student completing a 10 question test in 26 seconds is evidently not engaging with the module, and their inclusion will confound the results. Therefore, it is necessary to remove this data from the analysis.

Students were removed if the time it took them to complete the pre/post test was too long or too short. The standard approach to filtering based on time, which is to use $\mu \pm 2\sigma$ cutoffs, will not work because of the huge variance in completion times. Instead, we used a bottom end cutoff of 5 minutes. This was based on the amount of time it took me as a content expert and creator of the test to finish it. The upper end cutoff was 30 minutes, which was chosen because it was more than double the median values of 833 seconds and 929 second for the HLG pre-test and LLG pre-test (respectively). Applying these filters led to a combined (HLG and LLG) data set size of N=171. This filter also eliminated students who used the module more than once or appeared in both versions.

6.3.7 Classifying HPK/LPK Students

Multiple approaches for classifying students as HPK or LPK were considered. Among them, we implemented splitting these students according to their pretest scores according to the condition (HLG or LLG) median score. The median was chosen because scores on the pretest are discrete and using the average led to large imbalance between the knowledge classes for a given condition. In chapter 4 [4] we reported on the training and simulated deployment of a machine learning model that predicted student risk based on academic and behavioral features. We used this machine learning model to make predictions on un-labelled data available for students using the module. Although some students were automatically excluded from this analysis due to missing features in the dataset.

The final method we used to classify students was their responses to the cognitive load survey. HPK students should generally report lower ICL in both condtions (HLG and LLG) than their LPK counterparts. However, HPK students in the HLG version should report higher ECL than HPK students in the LLG condition. Conversely, LPK students should generally report higher ICL. LPK students in the LLG condition should report higher ECL than those in the HLG condition.

6.4 Results

6.4.1 Validation

In this section we present the results of the deployment of this module. We begin by demonstrating that the pre/post test used in this study is a valid instrument to assess student knowledge. To this end, we use *Kuder-Richardson Formula 20* (KR20). The KR20 metric prinipally examines the correctness-incorrectness rate and the variance of student answers. KR20 yields values that are in the range [0,1] where 0 is total un-reliability and 1 is total reliability [106].

$$KR20 = \frac{k}{k-1}\left(1 - \frac{\sum_{\mathrm{i}} p_{\mathrm{i}} q_{\mathrm{i}}}{\sigma^2}\right) \tag{6.1}$$

Where k is the number of questions, p_{i} is the fraction of students correctly solving each question, q_{i} is the fraction incorrectly solving the problem, and σ is the standard deviation of the total score. The results of KR20 are summarized in (tab. 6.3).

The results of KR20 indicate that the pre/post-test are very reliable instruments to assess student knowledge of forces and motion.

Table 6.3. KR20 scores for all students (HLG and LLG) for the pre-test and the post-test

Test	KR20
Pre-Test	0.96
Post-Test	0.98

6.4.2 HLG and LLG Scores

Next we present the test scores for the LLG condition not differentiated by prior knowledge level (HPK/LPK). All scores will be reported as $\mu \pm \frac{\sigma}{\sqrt{N}}$. The results are reported in tab. 6.4.

To provide further insight into the effect of the module on this group, we will examine the performance by question for pre-test and post-test in the LLG condition (fig. 6.2).

Table 6.4. Aggregate scores on pre and post test for the LLG condition out of 10 points

Test	Score
LLG-Pre	6.22 ± 0.18
LLG-Post	7.02 ± 0.20

We also present test scores for the HLG condition, not deliniated by prior knowledge

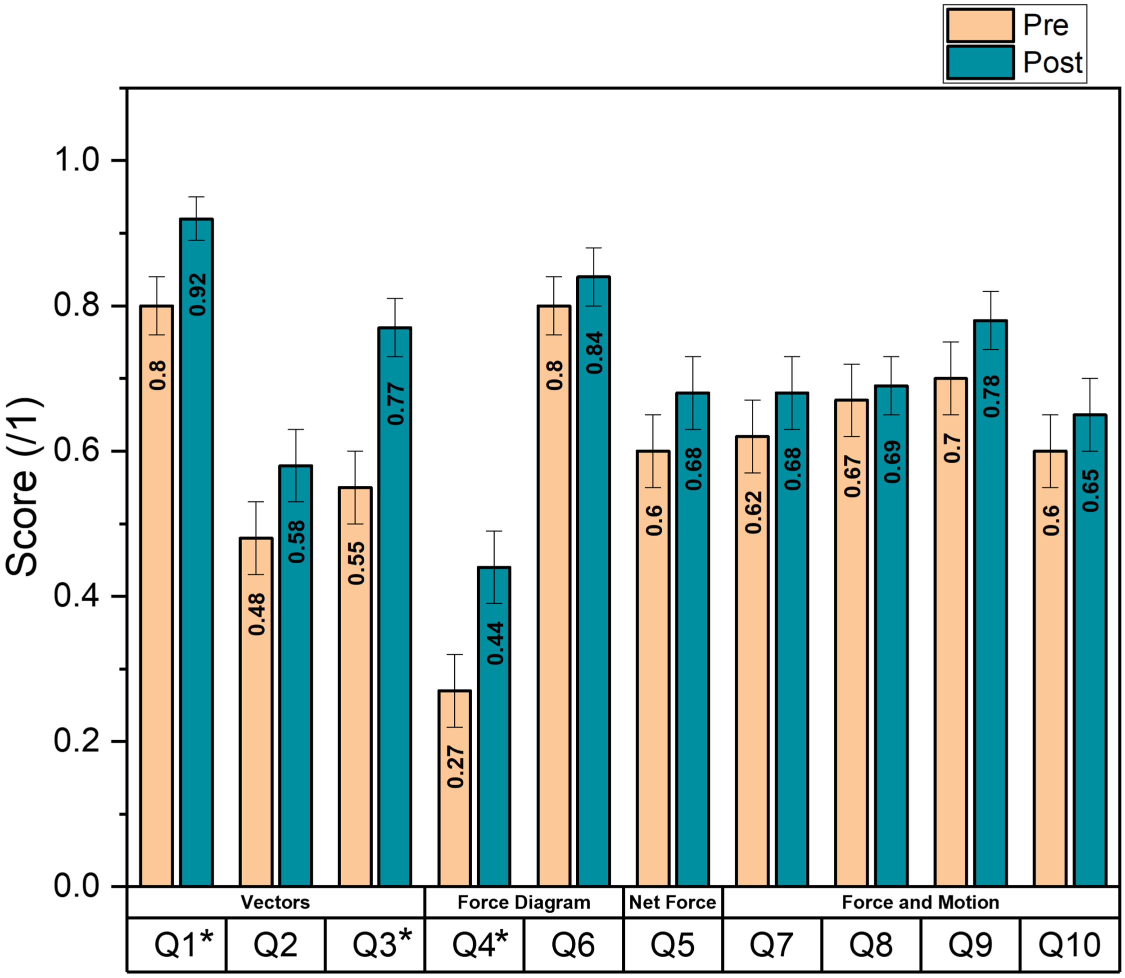


Figure 6.2. The graph shows a side-by-side comparison of the pre/post test performance of participants in the LLG condition. * indicates statistical significance at the level of $\alpha^* = 0.05$

level in tab. 6.5.

Now we present the performance by question for the pre/post test in the HLG condition

Table 6.5. Aggregate scores on pre and post test for the LLG condition out of 10 points

Test	Score
HLG-Pre	6.38 ± 0.20
HLG-Post	7.45 ± 0.18

(fig. 6.3).

We also report the results of the judgement of learning and cognitive load survey for

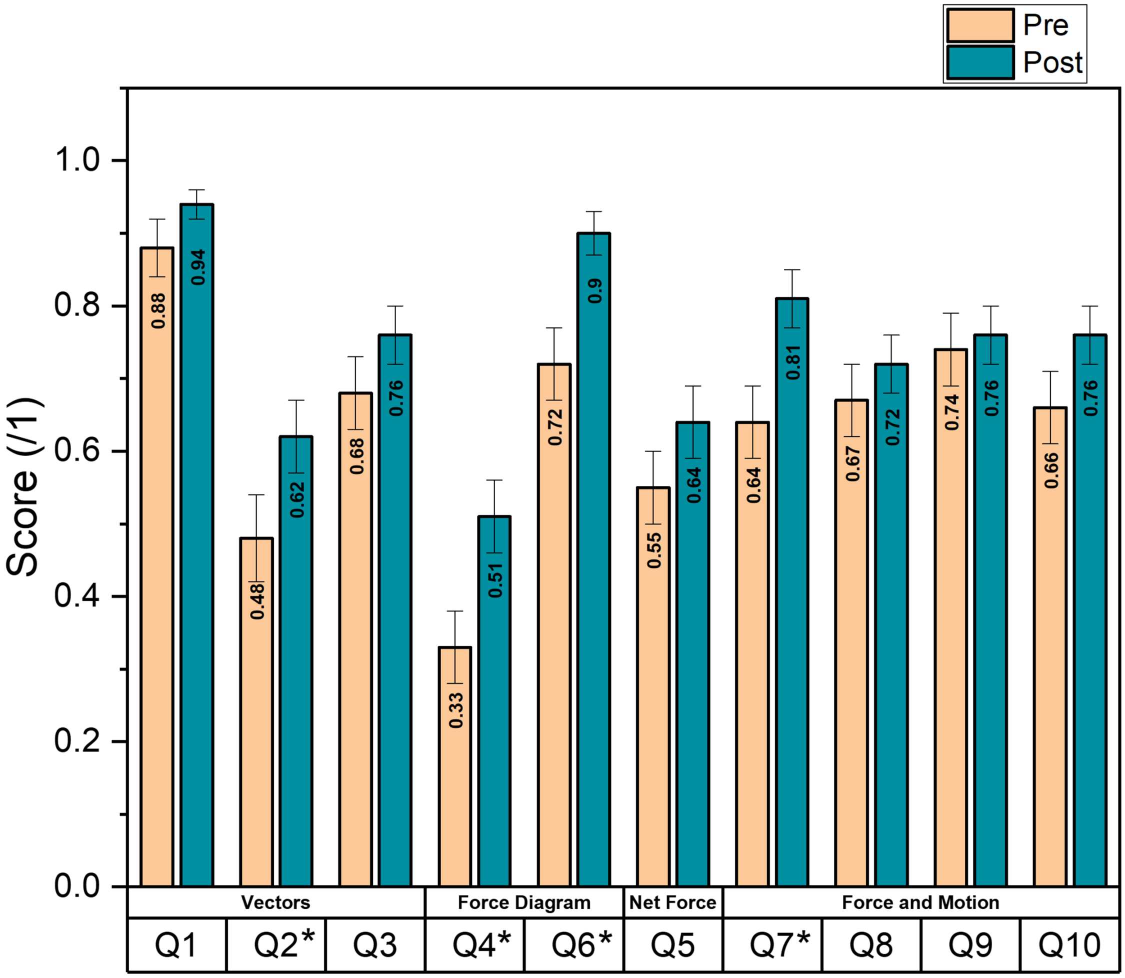


Figure 6.3. The graph shows a side-by-side comparison of the pre/post test performance of participants in the HLG condition. * indicates statistical significance at the level of $\alpha^* = 0.05$

the HLG and LLG participants.

Finally, we show the effect of the module as normalized gain[70] between pre-test and post-test. The normalized gain, also called the *Hake* gain uses the difference between the pre and post test scores, and is normalized by how much better participants *could have* done on the pre-test. The normalized gain is a useful metric, and criticisms that it is biased towards the pre-test score are unfounded [107].

Table 6.6. Results of the judgement of learning and cognitive load survey for students using the LLG version and the HLG version of the module. * denotes statistical significance at the level $\alpha^* = 0.05$

Item	HLG	LLG
Judgement of Learning*	$(81.56 \pm 1.35)\%$	$(72.18 \pm 1.82)\%$
Intrinsic Load	4.59 ± 0.19	5.01 ± 0.17
Extraneous Load	2.55 ± 0.16	2.84 ± 0.16
Germane Load*	6.89 ± 0.16	6.34 ± 0.17

$$\langle g \rangle = \frac{\langle post \rangle - \langle pre \rangle}{100\% - \langle pre \rangle} \tag{6.2}$$

Uncertainties were estimated using the rules of error propogation There was a statistically significant improvement for HLG and LLG from pre-test and post-test at the level $\alpha = 0.05$.

Table 6.7. Normalized gain ($\langle g \rangle$) by condition

Condition	$\langle g \rangle$
HLG	0.30 ± 0.07
LLG	0.21 ± 0.06

6.4.3 Classifying HPK/LPK Students

Now we report on classifying students as HPK and LPK. The XGBoost model trained in Chapter 4 was imported and used to make predictions. Specifically a model trained on academic and demographic features with the final course grade as the label, was used to make predictions on unlabelled data for students that completed the module. The predictions made by the model in Chapter 4 corresponded to risk of failing a physics course in which they were enrolled. Here, we used academic risk as a proxy for domain knowledge. Students predicted by the model as ’1’ (high-risk) were deemed LPK, and students predicted as ’0’ (low-risk) were deemed HPK.

The data for the HLG and LLG conditions were subsequently split into four groups (tab. 6.8): The definitions in tab. 6.8 led to the data being sorted in the following groups (tab. 6.9) The preformance of these groups are shown in figure 6.5

Table 6.8. Classes used for analysis of adaptive features of the module.

Group	Description
HLG-HPK	Classified as HPK and used HLG version
HLG-LPK	Classified as LPK and used HLG version
LLG-HPK	Classified as HPK and used LLG version
LLG-LPK	Classified as LPK and used LLG version

Table 6.9. Class Numbers Resulting from ML Sorting

Group	N
HLG-HPK	45
HLG-LPK	23
LLG-HPK	47
LLG-LPK	18

Another approach to HPK/LPK classification was to segment student groups by their score on the pretest. Students were grouped based on the HLG/LLG aggregate median on the pretest.

The final method for sorting students based on prior knowledge level was to use responses to the cognitive load survey. Students reporting at or above the median level of intrinsic cognitive load for their condition (HLG or LLG) were designated as LPK and students report ICL below the median value were designated as HPK.

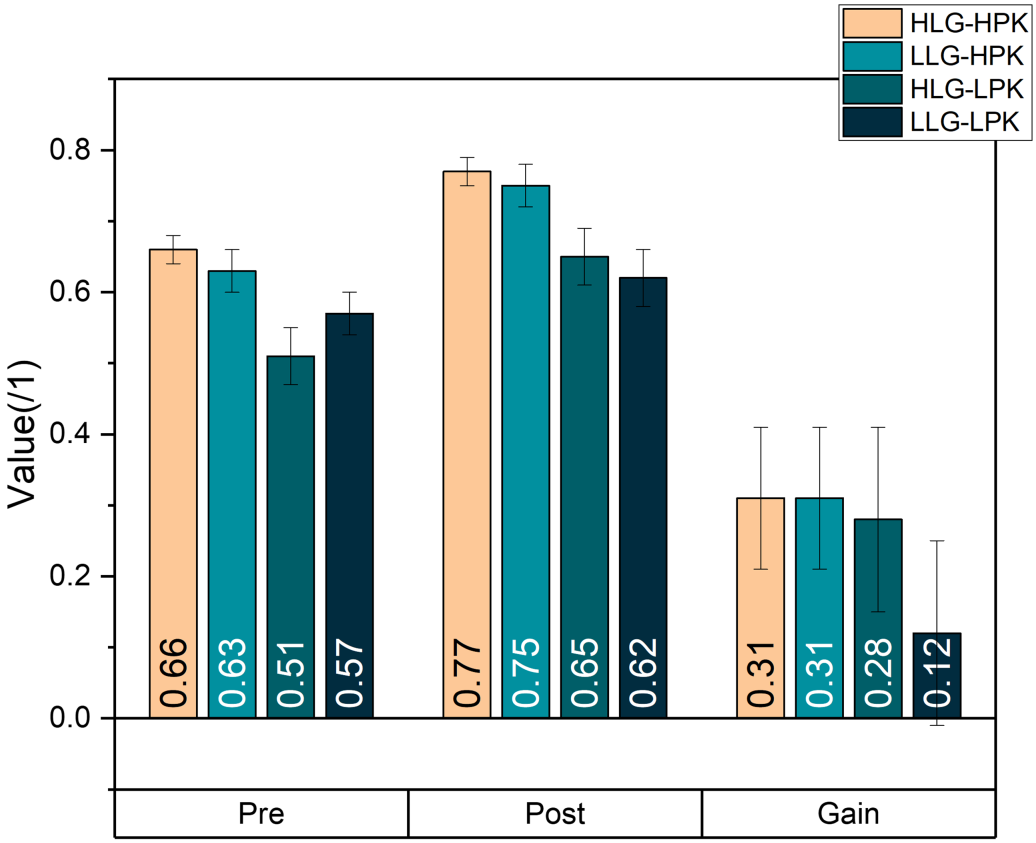


Figure 6.4. The pretest, post test scores are shown along with the normalized gain for the four groups sorted according to ML classification (tab. 6.9)

Table 6.10. Sorting criterion for HPK and LPK students from the two conditions HLG and LLG

Group	Criteria	N
HLG-HPK	$> \tilde{x}$	45
HLG-LPK	$\leq \tilde{x}$	37
LLG-HPK	$> \tilde{x}$	35
LLG-LPK	$\leq \tilde{x}$	49

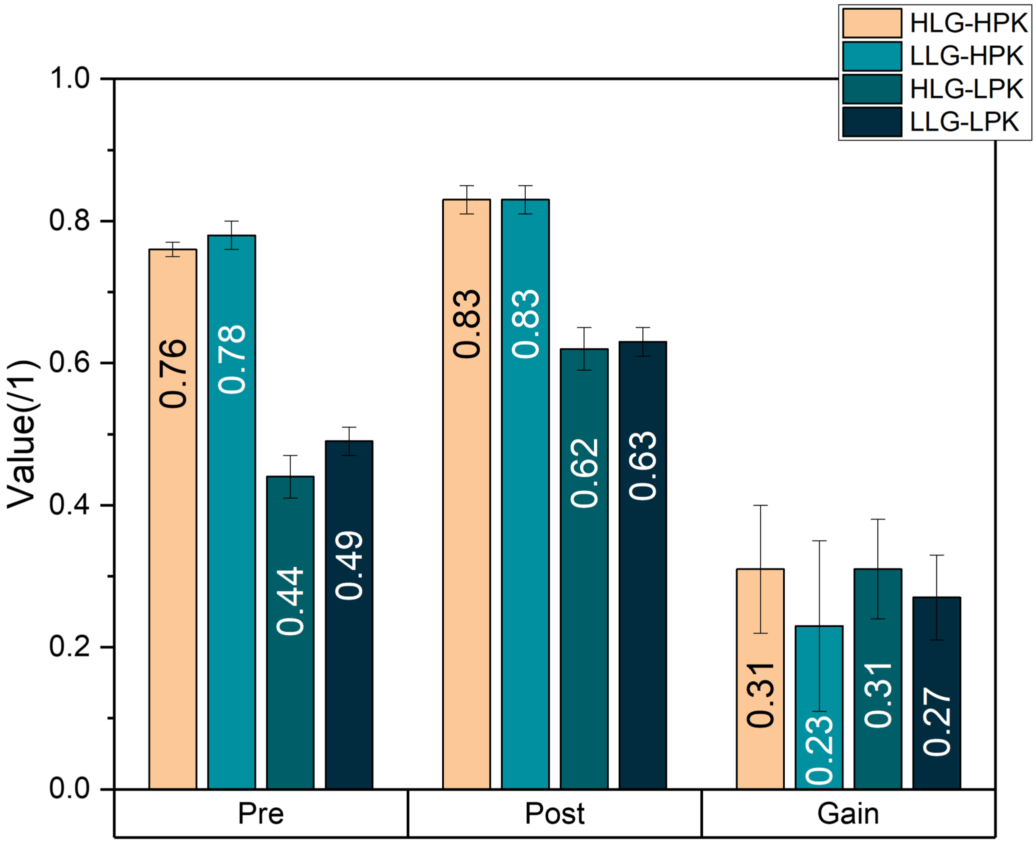


Figure 6.5. The pretest, post test scores are shown along with the normalized gain for the four groups sorted according to pretest score sorting (tab. 6.10)

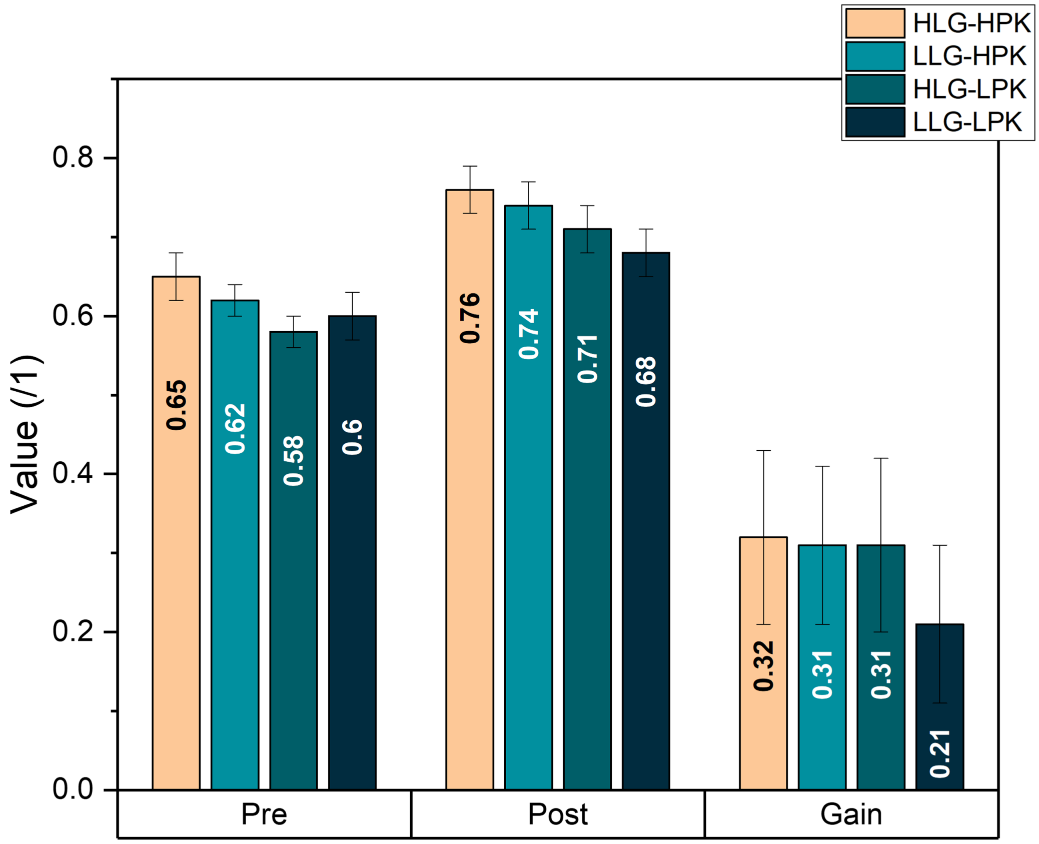


Figure 6.6. The pretest, post test scores are shown along with the normalized gain for the four groups sorted according to sorting by the cognitive load survey

6.5 Discussion

The results show a significant improvement for students using both the LLG and HLG versions of the module. However, the HLG version of the module led to a greater overall improvement. The question-by-question analysis shows that the modules are most effective at teaching students to reason with vectors and to use that skill to accurately create force diagrams, which are prerequisites to solving problems in force and motion. This analysis also shows that students had a strong initial ability to solve force and motion problems, althouth they were lacking in the fundamental skills of vector analysis and translating problem statements to force diagrams.

Students using the HLG version report significantly higher judgement of learning scores than in the LLG version. This reflects their perception that the HLG version is effective in imparting knowledge. Furthermore, participants using the HLG version reported a significantly higher GCL. Based on the literature, there are two possible interpretations of this result. According to cognitive load theory proper [3]–[5] this indicates that this version of the module is effective at facilitating schema development. According to Mayer [25] schema acquisition is the ultimate goal of multimedia instruction.

Another interpretation of high GCL relates to mental effort, such as ”concious application of a learning strategy” [108]. This result along with significantly higher judgement of learning scores indicate that students were more engaged as a result of using the HLG version of the module.

Our methods for grouping HPK and LPK students among participants who used the HLG and LLG versions of the module did not lead to any significant insights about the effectiveness of the different versions of the module for HPK and LPK students.

The machine learning approach was coherent in the sense that for HLG users the pretest scores for HLG-HPK were significantly higher than HLG-LPK as expected. The pretest scores for LLG-HPK were significantly different from LLG- LPK. Although, the expertise reversal effect was not demonstrated in the data.

Using the pretest scores to seperate HPK learners from LPK learners seems like a sound approach. The pretest scores were significantly different between these two groups. Granted that this is not an indepent metric to judge the suitablility of this method since the learners were split based on their pretest score. However, the pretest scores are significantly different between LPK and HPK which is not guarenteed by this method of sorting HPK and LPK learners. Also, the normalized gain hints at the expertise reversal effect without demonstrating it with statistical significance. The very large error bars associated with normalized gain prevent us from drawing any definite conclusions based on this metric alone.

Finally, The method of using the cognitive load survey along to split learners proves to be a very problematic approach. First, HPK and LPK learners show very similar pretest scores which is a strong indication that this method is invalid. Furthermore, according to Zu et. al. [105] LPK learners struggle to differentiate between different CL types in self-reported survey items. In light of these considerations, this method of differentiating LPK from HPK does not seem to be valid.

6.6 Conclusion

In this work we developed and deployed an instructional module. The module was used by students in a first-year algebra based Newtonian mechanics course. The module was randomly presented to students in one of two presentations. The HLG version used animation, narration, and explicit instruction to optimize instruction of LPK students. Meanwhile, the LLG version used pop-up content in place of animations and narration to reduce cognitive load for HPK learners.

The results showed that both versions of the module were effective in teaching principles of force and motion as evidence in an increase between the pretest and the post-test for all questions. Since the pretest and the post-test were indentical one might expect that in the absence of learning that had taken place as a result of using the module, students would default to using their answers from the pretest on the post-test. That didn't happen, instead there was a significant difference in the responses to these two instruments.

We conclude pursuant to RQ1 that we were able to successfully use online instructional modules to teach the principles of force and motion. This claim is based on the significant improvement between the post-test scores and the pretest scores.

The significant improvements shown between the pretest and the post-test as a result of using the module along with the instructional considerations outlined in section 6.2 indicate that we were able to optimize content in general for learners from this population as in RQ2.

RQ3 is related to the classification of HPK and LPK students. There is not a clear answer to this question at the present time. Both the approaches of using ML for classification and using pretest score are coherent in the sense that the pretest score is significantly different between HPK and LPK. The small sample size makes the error associated with performance metrics (particularly the normalized gain) such that no clear conclusions can be drawn. Therefore, we are forced to conclude that we were either not able to adequately sort HPK/LPK students or that the modules were not particularly well suited to provide adaptive instruction as in RQ4.

7. CONCLUSION

7.1 Overview

In this body of work we set forth with the goal to improve student learning in physics through automated differentiated feedback and instruction. We pursued this goal at three levels: At the macro level of the course as a whole i.e. in developing a model that would predict success in terms of final course grade, at the meso level of providing differentiated instruction through an online module within the course; and at the micro level of providing feedback on the quality of a written strategy essay on a single problem.

In chapter 4 we reported on the development of a machine learning algorithm (MLA) to predict students at risk of academic difficulty in a first year course in mechanics. Specifically, we used academic and demographic information to train a MLA to predict the final grade category of the student ('B-' or above [low-risk], and 'C+' and below[high-risk]). Our performance at predicting at-risk students was such that intervention could happen in a timely manner to prevent unfavorable outcomes for this population. In chapter 6, we tried to repurpose this algorithm to predict students' level of domain knowledge in physics (HPK or LPK). The results indicate that it may be a useful to categorize students in this way. However, due (in part) to the small numbers of students we were unable to draw any significant conclusions using this approach.

In chapter 5 we developed a machine learning algorithm to assess student scientific argumentation. We gave a quiz to future scientists and engineers where they were asked to solve a variation of the 'ballistic pendulum' problem. We codified the essays using text-vectorization methods and used the final (binary) score on the problem as a ground truth label. We had very good overall performance on this problem that was in line with commercial essay scoring platforms [92]. Specifically we note, that 84% of people who would go on to solve the problem incorrectly were predicted by the algorithm as likely to incorrectly solve the problem. Due to a significant number of errors made by the model, more work is needed on this problem before it can be used in real classrooms to provide useful interventions.

In chapter 6 we developed an online instructional module to provide instruction about the principles of force and motion. The content was delivered in two different styles that were intended to make the module adaptive to students based on their level of domain knowledge. The results showed that both versions of the module were effective at teaching the concepts of force and motion. However, we were unable to demonstrate a strong preference of low prior knowledge (LPK) learners for the module with a high level of guiding (HLG) features. Likewise for high prior knowledge (HPK) learners and the module with a low level of guiding features (LLG). This arrangement (HPK →LLG and LPK →HLG) is predicted by the expertise reversal effect (ERE) [12], [93]. Due to the level of error imposed by variation in learning outcomes and small sample sizes, we cannot make any conclusions about the ERE in multimedia physics instruction.

7.2 Future Work

7.2.1 Chapter 4

The performance of the model in chapter 4 indicates that more work is needed to improve the quality of predictions before it can be deployed in real life classrooms. One reason for lower than expected performance was missing features from the data. This problem cannot be directly addressed until record keeping improves. One potential way to overcome this obstacle is to obtain more data in terms of the number of features, and the number of students represented in the data. This would allow the use of advanced practices in filling missing features, such as training a machine learning model on adjacent features and using the model to predict the missing features.

Another way that performance could potentially be improved is to obtain raw data on student scores, rather than the adjusted final scores reported to the registrar. Using raw data would likely result in a better correlated set of features and labels.

Finally, it was shown in the literature [1], [2] that the inclusion of in-class features such as homework and exam scores, can improve the performace of the model.

7.2.2 Chapter 5

The results of a simulated deployment of the model in chapter 5 shows that this model needs to be refined to make better predictions. Since the model was trained on roughly the same number of samples that it was tested on, an obvious point is that more training data should greatly improve the prediction accuracy. Furthermore, in this work we used the problem correctness as the ground truth label. An interesting parallel is to score the essays on their own merit, and use the model to assess the quality of the scientific reasoning. It is a very labor intensive proposition, fraught with logistical challenges to hand score thousands of essays. Even so, this task is currently underway.

Subsequent work on this problem will be focused on improving prediction and generalization. In a series of studies that are planned for the future, the endpoints will correspond to notions of generalization. In phase 1 generalization, which we have already achieved to some degree, the trained model will predict data that it wasn't trained on (out of sample data) with a high degree of accuracy. In phase 2 generalization, the model will make high accuracy predictions on different problems from the same class of problems (momentum and energy conservation). In phase 3 generalization, the trained model will assess argumentation on any problem that is appropriate for this population.

The levels of generalization outlined in the previous paragraph will require very sophisticated methods. In chapter 5 we used very simple methods to represent text data, name bag-of-words. These high levels of generalization will require a model that is capable of understanding human writing. To this end, we will use the Bi-Directional Encoder Representations from Transformers (B.E.R.T.) model. The BERT model uses transformers with a built in *self attention* mechanism to learn contextualized word embeddings. While a detailed description of BERT is beyond the scope of this report, we can say that BERT is a huge advance in the endeavor of human language *understanding* by a machine. The use of BERT may be sufficient to detect patterns in scientific reasoning that lends itself to phase 3 generalization as defined above.

7.2.3 Chapter 6

In chapter 6 we were able to *generally* provide effective instruction to students on the principles of force and motion. However, we were unable to differentiate students based on their level of domain knowledge. Therefore, we were unable to make any conclusions about the benefit of the versions of the module to students based on domain knowledge. The method of sorting students based on their pretest score on the module hinted at the ERE, but failed to show it conclusively. The method of using a machine learning model trained on students' prior academic behaviors and demographic information was coherent in the sense that students predicted as HPK had higher pretest scores than those predicted as LPK.

Future work would necessarily include improving the ability to identify students' level of domain knowledge. Approaches to this problem may include using the pretest score along with behavioral data, and additional data such as the Force Concept Inventory score. With this data we could train a machine learning algorithm such as in chapter 4 to accurately predict student domain knowledge levels. With the ability to classify students' prior knowledge level we can revisit the design of the modules.

Currently, the HLG version of the module is being used on another study to detect patterns in student attention during online learning. In this project, students are asked to use the module in a lab setting while their eye movements are being logged along with a web cam that is providing a screen-forward view of the student during online learning. Students are then classified into one of four quadrants [109]   Quadrant 1 corresponds to a student who is looking at the lesson content and is actively thinking about it. Quadrant 2 is a student who is looking at the screen, but is engaged in thought not related to the lesson. A student in quadrant 3 is looking off screen and not thinking about the lesson. Meanwhile, quadrant

Table 7.1. Quadrant Descriptions Based on Gaze Target and Engagement

Quadrant	Looking at Screen	Engaged
1	yes	yes
2	yes	no
3	no	no
4	no	yes

4 is a student is looking off screen but is actively engaged in thought about the lesson.

One of the endpoints of this study is to produce software where the student uses the module while being observed by their own webcam outside of the lab. Data acquired by the webcam is then able to be used to classify the student into one of the four quadrants and intervention can happen in real time to get the student to re-engage with the lesson.

REFERENCES

[1] C. Zabriskie, J. Yang, S. DeVore, and J. Stewart, "Using machine learning to predict physics course outcomes," *Phys. Rev. Phys. Educ. Res.*, vol. 15, p. 020 120, 2 Sep. 2019. DOI: 10.1103/PhysRevPhysEducRes.15.020120. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.15.020120.

[2] J. Yang, S. DeVore, D. Hewagallage, P. Miller, Q. Ryan, and J. Stewart, "Using machine learning to identify the most at-risk students in physics classes," *Phys. Rev. Phys. Educ. Res.*, vol. 16, p. 020 130, 2 Oct. 2020. DOI: 10.1103/PhysRevPhysEducRes.16.020130. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.16.020130.

[3] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, pp. 257–285, 2 1988.

[4] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Educational Psychology Review*, vol. 4, pp. 295–312, 4 1994.

[5] J. Sweller, "Element interactivity and intrinsic, extraneous, and germane cognitive load," *Educational Psychology Review*, vol. 22, pp. 123–138, 2010.

[6] J. Docktor and J. Mestre, "Synthesis of discipline-based education research in physics," *Phys. Rev. ST Phys. Educ. Res.*, vol. 10, p. 020 119, 2 Sep. 2014. DOI: 10.1103/PhysRevSTPER.10.020119. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevSTPER.10.020119.

[7] D. Maloney, "An overview of physics education research on problem solving," in *Getting Started in PER*, 1st ed., vol. 2, Sep. 2011.

[8] R. Atkinson, A. Renkl, and M. Merrill, "Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps," English (US), *Journal of Educational Psychology*, vol. 95, no. 4, pp. 774–783, Dec. 2003, ISSN: 0022-0663. DOI: 10.1037/0022-0663.95.4.774.

[9] R. E. Mayer, C. C. Stiehl, and J. G. Greeno, "Acquisition of understanding and skill in relation to subjects' preparation and meaningfulness of instruction," *Journal of Educational Psychology*, vol. 67, pp. 331–350, 3 1975. DOI: https://doi.org/10.1037/h0076619.

[10] A. Ollerenshaw, E. Aidman, and G. Kidd, "Is an illustration always worth ten thousand words? effects of prior knowledge, learning style and multimedia illustrations on text comprehension," *International Journal of Instructional Media*, vol. 22, Jan. 1997.

[11] E. Pollock, P. Chandler, and J. Sweller, "Assimilating complex information," *Learning and Instruction*, vol. 12, no. 1, pp. 61–86, 2002, ISSN: 0959-4752. DOI: https://doi.org/10.1016/S0959-4752(01)00016-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0959475201000160.

[12] S. Kalyuga, P. Chandler, and J. Sweller, "Incorporating learner experience into the design of multimedia instruction.," *Journal of Educational Psychology*, vol. 92, pp. 126–136, 2000.

[13] R. Saadé, D. Morin, and J. Thomas, "Critical thinking in e-learning environments," *Comput. Hum. Behav.*, vol. 28, pp. 1608–1617, 2012.

[14] A. Baddeley and G. Hitch, "Working memory," in ser. Psychology of Learning and Motivation, G. H. Bower, Ed., vol. 8, Academic Press, 1974, pp. 47–89. DOI: https://doi.org/10.1016/S0079-7421(08)60452-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079742108604521.

[15] A. Baddeley, "The episodic buffer: A new component of working memory?" *Trends in Cognitive Sciences*, vol. 4, no. 11, pp. 417–423, 2000, ISSN: 1364-6613. DOI: https://doi.org/10.1016/S1364-6613(00)01538-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661300015382.

[16] M. Chi, P. Feltovich, and R. Glaser, "Categorization and representation of physics problems by experts and novices*," *Cognitive Science*, vol. 5, no. 2, pp. 121–152, 1981. DOI: https://doi.org/10.1207/s15516709cog0502_2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog0502_2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0502_2.

[17] P. Chandler and J. Sweller, "Cognitive load theory and the format of instruction," *Faculty of Education - Papers*, vol. 8, Dec. 1991. DOI: 10.1207/s1532690xci0804_2.

[18] T. Zu, "Using multiple ways to investigate cognitive load theory in the context of physics instruction," Ph.D. dissertation, Purdue University, 2012.

[19] J. Sweller, P. Ayres, and S. Kalyuga, "The element interactivity effect," in *Cognitive Load Theory*. New York, NY: Springer New York, 2011, pp. 193–201, ISBN: 978-1-4419-8126-4. DOI: 10.1007/978-1-4419-8126-4_15. [Online]. Available: https://doi.org/10.1007/978-1-4419-8126-4_15.

[20] P. Ayres, "Why goal-free problems can facilitate learning," *Contemporary Educational Psychology*, vol. 18, pp. 376–381, 3 1993. DOI: 10.1006/ceps.1993.1027. [Online]. Available: https://doi.org/10.1006/ceps.1993.1027.

[21] A. Renkl, "The worked-out examples principle in multimedia learning," in *The Cambridge Handbook of Multimedia Learning*, R. Mayer, Ed., ser. Cambridge Handbooks in Psychology. Cambridge University Press, 2005, pp. 229–246. DOI: 10.1017/CBO9780511816819.016.

[22] O. Chen, S. Kalyuga, and J. Sweller, "The expertise reversal effect is a variant of the more general element interactivity effect," *Educational Psychololgy Review*, vol. 29, pp. 393–405, 2017. DOI: 10.1007/s10648-016-9359-1. [Online]. Available: https://doi.org/10.1007/s10648-016-9359-1.

[23] L. Cronbach, "The two disciplines of scientific psychology," *American Psychologist*, vol. 12, no. 1, pp. 671–684, 11 1957. DOI: https://doi.org/10.1037/h0043943.

[24] D. Lohman, "Predicting mathemathanic effects in the teaching of higher-order thinking skills," *Educational Psychologist*, vol. 21, pp. 191–208, 3 Jun. 1986. DOI: 10.1207/s15326985ep2103_3.

[25] R. Mayer, K. Steinhoff, and G. e. a. Bower, "A generative theory of textbook design: Using annotated illustrations to foster meaningful learning of science text," *ETR&D*, vol. 43, pp. 31–41, Mar. 1995. DOI: 10.1007/BF02300480. [Online]. Available: https://doi.org/10.1007/BF02300480.

[26] A. Brunstein, S. Betts, and J. Anderson, "Practice enables successful learning under minimal guidance," *Journal of Educational Psychology*, vol. 101, pp. 790–802, 4 2009. DOI: https://doi.org/10.1037/a0016656.

[27] J. P. Mestre, J. L. Docktor, N. E. Strand, and B. H. Ross, "Chapter nine - conceptual problem solving in physics," in ser. Psychology of Learning and Motivation, J. Mestre and B. Ross, Eds., vol. 55, Academic Press, 2011, pp. 269–298. DOI: https://doi.org/10.1016/B978-0-12-387691-1.00009-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123876911000090.

[28] M. Rosenberg and R. Foshay, "E-learning: Strategies for delivering knowledge in the digital age," *Performance Improvement*, vol. 41, no. 5, pp. 50–51, 2002. DOI: https://doi.org/10.1002/pfi.4140410512. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pfi.4140410512. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/pfi.4140410512.

[29] F. Keller, ""good-bye, teacher..."1," *J Appl Behav Anal.*, vol. 1, no. 1, pp. 79–89, Mar. 1968. DOI: doi:10.1901/jaba.1968.1-79. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1310979/.

[30] F. Moedritscher, V. Garcia-Barrios, and C. Guetl, “C.: “the past, the present and the future of adaptive e-learning: An approach within the scope of the research project adele,” Jan. 2004. [Online]. Available: https://www.researchgate.net/publication/242378532_C_The_Past_the_Present_and_the_Future_of_adaptive_E-Learning_An_Approach_within_the_Scope_of_the_Research_Project_AdeLE.

[31] D. Burgos, C. Tattersall, and R. Koper, “Representing adaptive elearning strategies in ims learning design,” 2006.

[32] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[33] S. Chan, *Ece 595 lecture notes*, Jan. 2018.

[34] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2008.

[35] I. Rish, “An empirical study of the naïve bayes classifier,” *IJCAI 2001 Work Empir Methods Artif Intell*, vol. 3, Jan. 2001.

[36] V. Vapnik and A. Lerner, “Pattern recognition using generalized portrait method,” *Automation and Remote Control*, vol. 24, pp. 774–780, 1963.

[37] T. Murphy, P. Gray, and G. Stewart, *Certified convergent perceptron learning*, 1960.

[38] R. Jain, *Decision tree. it begins here.* 2017. [Online]. Available: https://medium.com/@rishabhjain%2022692/decision-trees-it-begins-here-93ff54ef134.

[39] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[40] T. Winograd, “Understanding natural language,” *Cognitive Psychology*, vol. 3, no. 1, pp. 1–191, 1972, ISSN: 0010-0285. DOI: https://doi.org/10.1016/0010-0285(72)90002-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010028572900023.

[41] N. R. Council, *Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century*, J. Pellegrino and M. Hilton, Eds. Washington, DC: The National Academies Press, 2012, ISBN: 978-0-309-25649-0. DOI: 10.17226/13398. [Online]. Available: https://www.nap.edu/catalog/13398/education-for-life-and-work-developing-transferable-knowledge-and-skills.

[42] N. R. Council, *Developing Assessments for the Next Generation Science Standards*, J. Pellegrino, M. Wilson, J. Koenig, and A. Beatty, Eds. Washington, DC: The National Academies Press, 2014, ISBN: 978-0-309-28951-1. DOI: 10.17226/18409. [Online]. Available: https://www.nap.edu/catalog/18409/developing-assessments-for-the-next-generation-science-standards.

[43] C. Harris, J. Krajcik, J. Pellegrino, and A. DeBarger, "Designing knowledge-in-use assessments to promote deeper learning," *Educational Measurement: Issues and Practice*, vol. 38, no. 2, pp. 53–67, 2019. DOI: https://doi.org/10.1111/emip.12253. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/emip.12253. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/emip.12253.

[44] L. Kaldaras, H. Akaeze, and J. Krajcik, "Developing and validating next generation science standards-aligned learning progression to track three-dimensional learning of electrical interactions in high school physical science," *Journal of Research in Science Teaching*, vol. 58, no. 4, pp. 589–618, 2021. DOI: https://doi.org/10.1002/tea.21672. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/tea.21672. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/tea.21672.

[45] X. Zhai, "Applying machine learning in science assessment: Opportunity and challenge," *Journal of Science Education and Technology*, 2019.

[46] S. Maestrales, X. Zhai, I. Touitou, Q. Baker, B. Schneider, and J. Krajcik, "Using machine learning to score multi-dimensional assessments of chemistry and physics," *Journal of Science Education and Technology*, vol. 30, Apr. 2021. DOI: 10.1007/s10956-020-09895-9.

[47] T. Sun, C. Wang, R. Lambert, and L. Liu, "Relationship between second language english writing self-efficacy and achievement: A meta-regression analysis," *Journal of Second Language Writing*, vol. 53, p. 100 817, 2021, ISSN: 1060-3743. DOI: https://doi.org/10.1016/j.jslw.2021.100817. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1060374321000321.

[48] X. Zhai, Y. Yin, J. Pellegrino, K. Haudek, and L. Shi, "Applying machine learning in science assessment: A systematic review," *Studies in Science Education*, vol. 56, no. 1, pp. 111–151, 2020.

[49] O. L. Liu, J. A. Rios, M. Heilman, L. Gerard, and M. C. Linn, "Validation of automated scoring of science assessments," *Journal of Research in Science Teaching*, vol. 53, no. 2, pp. 215–233, 2016. DOI: https://doi.org/10.1002/tea.21299. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/tea.21299. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/tea.21299.

[50] M. Ha and R. Nehm, "The impact of misspelled words on automated computer scoring: A case study of scientific explanations," *Journal of Science Education and Technology*, vol. 25, Jun. 2016. DOI: 10.1007/s10956-015-9598-9.

[51] C. Nakamura, S. Murphy, M. Christel, S. Stevens, and D. Zollman, "Automated analysis of short responses in an interactive synthetic tutoring system for introductory physics," *Phys. Rev. Phys. Educ. Res.*, vol. 12, p. 010 122, 1 Mar. 2016. DOI: 10.1103/PhysRevPhysEducRes.12.010122. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.12.010122.

[52] I. Okoye, S. Bethard, and T. Sumner, "CU : Computational assessment of short free text answers - a tool for evaluating students' understanding," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 603–607. [Online]. Available: https://aclanthology.org/S13-2101.

[53] Y. Kim, *Convolutional neural networks for sentence classification*, 2014. arXiv: `1408.5882 [cs.CL]`.

[54] R. E. Bennett, "Educational assessment: What to watch in a rapidly changing world," *Educational Measurement: Issues and Practice*, vol. 37, no. 4, pp. 7–15, 2018. DOI: https://doi.org/10.1111/emip.12231. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/emip.12231. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/emip.12231.

[55] "An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts," *User Modeling and User-Adapted Interaction*, vol. 21, pp. 99–135, 1-2 2011.

[56] F. Zehner, F. Goldhammer, and C. Sälzer, "Automatically analyzing text responses for exploring gender-specific cognitions in pisa reading," *Large-scale Assess Educ.*, vol. 6, 7. DOI: https://doi.org/10.1186/s40536-018-0060-3.

[57] J. Rosenberg and C. Krist, "Combining machine learning and qualitative methods to elaborate students' ideas about the generality of their model-based explanations," *Journal of Science Education and Technology*, vol. 30, pp. 1–13, Apr. 2021. DOI: 10.1007/s10956-020-09862-4.

[58] S. Sung, C. Li, G. Chen, *et al.*, "How does augmented observation facilitate multimodal representational thinking? applying deep learning to decode complex student construct," *Journal of Science Education and Technology*, vol. 30, no. 2, pp. 210–226, 2021.

[59] R. Mason and M. Just, "Neural representations of physics concepts," *Psychological Science*, vol. 27, no. 6, pp. 904–913, 2016, PMID: 27113732. DOI: 10.1177/0956797616641941. eprint: https://doi.org/10.1177/0956797616641941. [Online]. Available: https://doi.org/10.1177/0956797616641941.

[60] B. R., F. S.J., and N. R.H., "Enhancing data pipelines for forecasting student performance: Integrating feature selection with cross-validation.," *Int J Educ Technol High Educ.*, vol. 18, 1 2021. DOI: doi:10.1186/s41239-021-00279-6.

[61] L. Jescovitch, E. Scott, and J. e. a. Cerchiara, "Comparison of machine learning performance using analytic and holistic coding approaches across constructed response assessments aligned to a science learning progression.," *Sci Educ. Technol.*, vol. 30, pp. 150–167, 2021. DOI: https://doi.org/10.1007/s10956-020-09858-0.

[62] H. Liaw, Y. Yu, C. Chou, and M. Chiu, "Relationships between facial expressions, prior knowledge, and multiple representations: A case of conceptual change for kinematics instruction," *Journal of Science Education and Technology*, vol. 30, pp. 1–12, Apr. 2021. DOI: 10.1007/s10956-020-09863-3.

[63] H. Lee, A. Pallant, S. Pryputniewicz, T. Lord, M. Mulholland, and O. Liu, "Automated text scoring and real-time adjustable feedback: Supporting revision of scientific arguments involving uncertainty," *Science Education*, vol. 103, no. 3, pp. 590–622, 2019. DOI: https://doi.org/10.1002/sce.21504. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sce.21504. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sce.21504.

[64] L. Gerard and M. Linn, "Using automated scores of student essays to support teacher guidance in classroom inquiry," *Journal of Science Teacher Education*, vol. 27, Mar. 2016. DOI: 10.1007/s10972-016-9455-6.

[65] L. Vitale, E. McBride, and M. Linn, "Distinguishing complex ideas about climate change: Knowledge integration vs. specific guidance," *International Journal of Science Education*, vol. 38, no. 9, pp. 1548–1569, 2016. DOI: 10.1080/09500693.2016.1198969. eprint: https://doi.org/10.1080/09500693.2016.1198969. [Online]. Available: https://doi.org/10.1080/09500693.2016.1198969.

[66] L. Gerard, A. Kidron, and M. Linn, "Guiding collaborative revision of science explanations," *International Journal of Computer-Supported Collaborative Learning*, pp. 1–34, 2019.

[67] X. Zhai and M. Zhang, "One-to-one mobile technology in high school physics classrooms: Understanding its use and outcome," *British Journal of Educational Technology*, vol. 49, pp. 516–532, Dec. 2018. DOI: 10.1111/bjet.12539.

[68] J. Krajcik, "Commentary—applying machine learning in science assessment: Opportunity and challenges.," *J. Sci. Educ. Technol.*, vol. 30, pp. 313–318, 2021. DOI: https://doi.org/10.1007/s10956-021-09902-7.

[69] S. Cotner and C. Ballen, "Can mixed assessment methods make biology classes more equitable?" *PLoS ONE*, vol. 12, 12 2017. DOI: https://doi.org/10.1371/journal.pone.0189610.

[70] R. Hake, "Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *American Journal of Physics*, vol. 66, no. 1, pp. 64–74, 1998. DOI: 10.1119/1.18809. eprint: https://doi.org/10.1119/1.18809. [Online]. Available: https://doi.org/10.1119/1.18809.

[71] J. Leppink, F. Paas, C. Van der Vleuten, and et al, "Development of an instrument for measuring different types of cognitive load.," *Behav. Res.*, vol. 45, pp. 1058–1072, 2013. DOI: https://doi.org/10.3758/s13428-013-0334-1.

[72] *U.s. department of education | america's strategy for stem education [online]*. [Online]. Available: https://www.ed.gov/stem#background.

[73] X. Chen, "Stem attrition: College students' paths into and out of stem fields," Tech. Rep., Nov. 2013.

[74] R. Darolia, C. Koedel, J. Main, J. Ndashimye, and J. Yan, "High school course access and postsecondary stem enrollment and attainment," *Educational Evaluation and Policy Analysis*, vol. 42, no. 1, pp. 22–45, 2020. DOI: 10.3102/0162373719876923. eprint: https://doi.org/10.3102/0162373719876923. [Online]. Available: https://doi.org/10.3102/0162373719876923.

[75] B. Kennedy, R. Fry, and C. Funk, *6 facts about america's stem workforce and those training for it*, Apr. 2021. [Online]. Available: https://www.pewresearch.org/fact-tank/2021/04/14/6-facts-about-americas-stem-workforce-and-those-training-for-it/.

[76] Y. Xue and R. Larson, *Stem crisis or stem surplus? yes and yes : Monthly labor review*, 2015. [Online]. Available: https://www.bls.gov/opub/mlr/2015/article/stem-crisis-or-stem-surplus-yes-and-yes.htm.

[77] D. Beede, T. Julian, D. Langdon, B. McKittrick, and M. Doms, *Women in stem: A gender gap to innovation*, 2010.

[78] R. Fry, B. Kennedy, and C. Funk, *Stem jobs see uneven progress in increasing gender, racial and ethnic diversity*, Apr. 2021. [Online]. Available: https://www.pewresearch.org/science/2021/04/01/stem-jobs-see-uneven-progress-in-increasing-gender-racial-and-ethnic-diversity/.

[79] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. DOI: 10.1109/TKDE.2008.239.

[80] R. Dufresne, W. Gerace, P. Hardiman, and J. Mestre, "Constraining novices to perform expertlike problem analyses: Effects on schema acquisition," *The Journal of the Learning Sciences*, vol. 2, no. 3, pp. 307–331, 1992, ISSN: 10508406, 15327809. [Online]. Available: http://www.jstor.org/stable/1466611.

[81] W. Leonard, R. Dufresne, and J. Mestre, "Using qualitative problem-solving strategies to highlight the role of conceptual knowledge in solving problems," *American Journal of Physics*, vol. 64, no. 12, pp. 1495–1503, 1996. DOI: 10.1119/1.18409. eprint: https://doi.org/10.1119/1.18409. [Online]. Available: https://doi.org/10.1119/1.18409.

[82] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[83] *Random Decision Forest*, (Montreal, Canada), 1995, pp. 278–282.

[84] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 785–794, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785.

[85] S. Bird, L. Edward, and K. Ewan, *Natural Language Processing with Python.* O'Reilly Media Inc., 2009.

[86] S. Loria, *Textblob documentation. release 0.15, 2.* 2018.

[87] J. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, vol. 27, no. 3, pp. 221–234, 1987, ISSN: 0020-7373. DOI: https://doi.org/10.1016/S0020-7373(87)80053-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020737387800536.

[88] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning : with Applications in R.* New York: Springer, 2013.

[89] J. Freidman, "Greedy function approximation:a gradient boosting machine," vol. 29, 2001, pp. 1189–1232.

[90] M. McHugh, "Interrater reliability: The kappa statistic.," *Biochem Med*, vol. 22, pp. 276–282, 3 2012.

[91] G. Chung and E. Baker, “Automated essay scoring: A cross-disciplinary perspective.,” in *Issues in the Reliability and Validity of Automated Scoring of Constructed Responses*, New Jersey: Lawrence Erlbaum Associates, 2003.

[92] M. Shermis, “State-of-the-art automated essay scoring: Competition, results, and future directions from a united states demonstration,” *Assessing Writing*, vol. 20, pp. 53–76, 2014.

[93] S. Kalyuga, P. Chandler, and J. Sweller, “Learner experience and efficiency of instructional guidance,” *Educational Psychology*, vol. 21, no. 1, pp. 5–23, 2001. DOI: 10.1080/01443410124681. eprint: https://doi.org/10.1080/01443410124681. [Online]. Available: https://doi.org/10.1080/01443410124681.

[94] J. Sweller, P. Ayres, and S. Kalyuga, “The expertise reversal effect,” in *Cognitive Load Theory*. New York, NY: Springer New York, 2011, pp. 155–170, ISBN: 978-1-4419-8126-4. DOI: 10.1007/978-1-4419-8126-4_12. [Online]. Available: https://doi.org/10.1007/978-1-4419-8126-4_12.

[95] S. Dhawan, “Online learning: A panacea in the time of covid-19 crisis,” *Journal of Educational Technology Systems*, vol. 49, no. 1, pp. 5–22, 2020. DOI: 10.1177/0047239520934018. eprint: https://doi.org/10.1177/0047239520934018. [Online]. Available: https://doi.org/10.1177/0047239520934018.

[96] J. Paul and F. Jefferson, “A comparative analysis of student performance in an online vs. face-to-face environmental science course from 2009 to 2016,” *Frontiers in Computer Science*, vol. 1, 2019, ISSN: 2624-9898. DOI: 10.3389/fcomp.2019.00007. [Online]. Available: https://www.frontiersin.org/article/10.3389/fcomp.2019.00007.

[97] R. Mcdaniel, *Online course module structure*. [Online]. Available: https://www.vanderbilt.edu/cdr/module1/online-course-module-structure/.

[98] R. Gopal, V. Singh, and A. Aggarwal, “Impact of online classes on the satisfaction and performance of students during the pandemic period of covid 19,” *Educ Inf Technol*, vol. 26, pp. 6923–6947, 2021. DOI: https://doi.org/10.1007/s10639-021-10523-1.

[99] R. Mayer, “Multimedia learning,” in ser. Psychology of Learning and Motivation, vol. 41, Academic Press, 2002, pp. 85–139. DOI: https://doi.org/10.1016/S0079-7421(02)80005-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079742102800056.

[100] *12 principles of multimedia - web.home.syr.nyu.edu*. [Online]. Available: https://web.home.syr.nyu.edu/content/dam/nyu/facultyResources/documents/ESMITS/12PrinciplesofMultimedia.pdf.

[101] J. Richter and K. S., "Studying the expertise reversal of the multimedia signaling effect at a process level: Evidence from eye tracking," *Instructional Science*, vol. 47, pp. 627–658, 6. DOI: https://doi.org/http://dx.doi.org/10.1007/s11251-019-09492-3.

[102] J. Sweller, P. Ayres, and S. Kalyuga, "The modality effect," in *Cognitive Load Theory*. New York, NY: Springer New York, 2011, pp. 129–140, ISBN: 978-1-4419-8126-4. DOI: 10.1007/978-1-4419-8126-4_10. [Online]. Available: https://doi.org/10.1007/978-1-4419-8126-4_10.

[103] J. Sweller, P. Ayres, and S. Kalyuga, "The split-attention effect," in *Cognitive Load Theory*. New York, NY: Springer New York, 2011, pp. 111–128, ISBN: 978-1-4419-8126-4. DOI: 10.1007/978-1-4419-8126-4_9. [Online]. Available: https://doi.org/10.1007/978-1-4419-8126-4_9.

[104] J. Sweller, P. Ayres, and S. Kalyuga, "Emerging themes in cognitive load theory: The transient information and the collective working memory effects," in *Cognitive Load Theory*. New York, NY: Springer New York, 2011, pp. 219–233, ISBN: 978-1-4419-8126-4. DOI: 10.1007/978-1-4419-8126-4_17. [Online]. Available: https://doi.org/10.1007/978-1-4419-8126-4_17.

[105] T. Zu, J. Munsell, and N. Rebello, "Subjective measure of cognitive load depends on participants' content knowledge level," *Frontiers in Education*, vol. 6, 2021, ISSN: 2504-284X. DOI: 10.3389/feduc.2021.647097. [Online]. Available: https://www.frontiersin.org/article/10.3389/feduc.2021.647097.

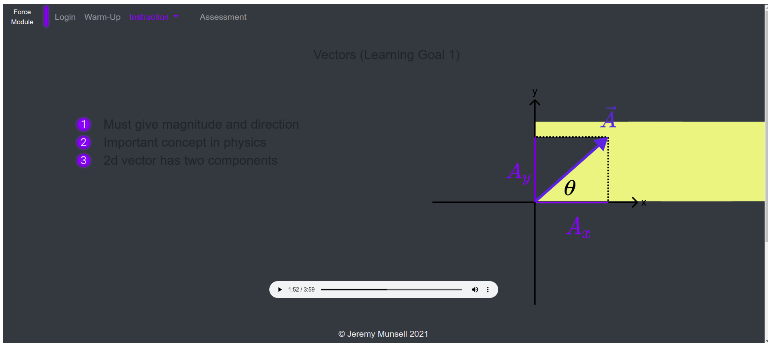
[106] *Encyclopedia of Research Design*, vol. 1, SAGE Publications, Inc, 2010. DOI: 10.4135/9781412961288. [Online]. Available: https://methods.sagepub.com/reference/encyc-of-research-design.

[107] V. Coletta and J. Steinert, "Why normalized gain should continue to be used in analyzing preinstruction and postinstruction scores on concept inventories," *Phys. Rev. Phys. Educ. Res.*, vol. 16, p. 010 108, Oct. 2020.

[108] N. Debue and C. van de Leemput, "What does germane load mean? an empirical contribution to the cognitive load theory," *Frontiers in Psychology*, vol. 5, 2014, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2014.01099. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2014.01099.

[109] S. D'Mello, "Giving eyesight to the blind: Towards attention-aware aied," *International Journal of Artificial Intelligence in Education*, vol. 26, pp. 645–659, 2 2016.

A. APPENDIX A.1 IMAGES FROM THE MODULES IN STUDY 3

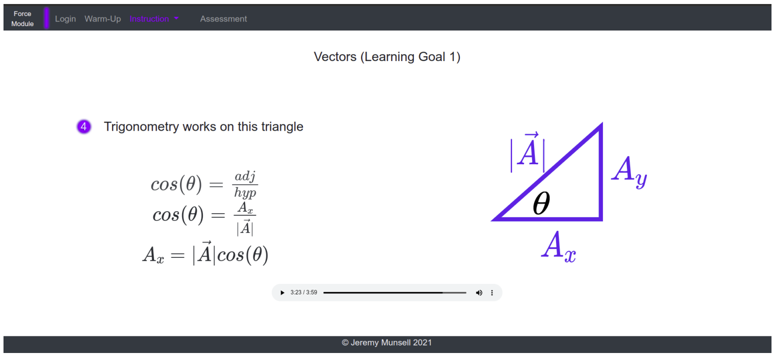


Vectors (Learning Goal 1)

1. Must give magnitude and direction
2. Important concept in physics
3. 2d vector has two components

© Jeremy Munsell 2021

Figure A.1. An analogy is presented where the x and y components of the vector are like the shadow of the vector on x and y axis (respectively) in the HLG version (6.3.1).



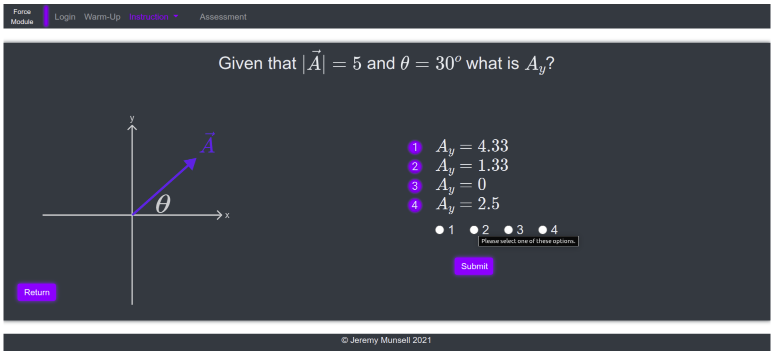
Vectors (Learning Goal 1)

4. Trigonometry works on this triangle

$$cos(\theta) = \frac{adj}{hyp}$$
$$cos(\theta) = \frac{A_x}{|\vec{A}|}$$
$$A_x = |\vec{A}|cos(\theta)$$

© Jeremy Munsell 2021

Figure A.2. The video shows how trigonometry can be applied to the right triangle formed by a vector and its components in the HLG version (6.3.1).

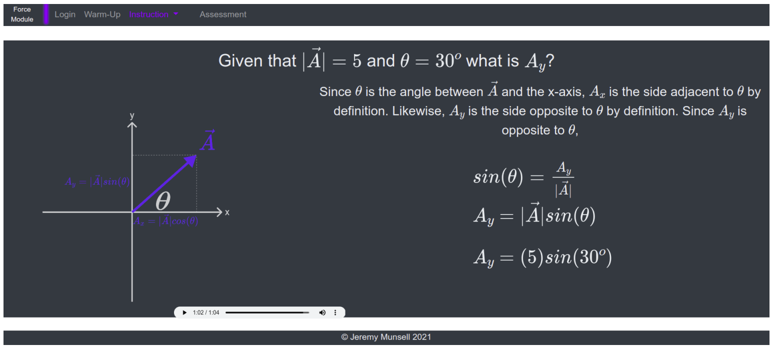


Given that $|\vec{A}| = 5$ and $\theta = 30^o$ what is A_y?

1. $A_y = 4.33$
2. $A_y = 1.33$
3. $A_y = 0$
4. $A_y = 2.5$

Please select one of these options.

Figure A.3. The assessment question given at the end of the vectors subsection in the HLG and LLG version (6.3.1).



Given that $|\vec{A}| = 5$ and $\theta = 30^o$ what is A_y?

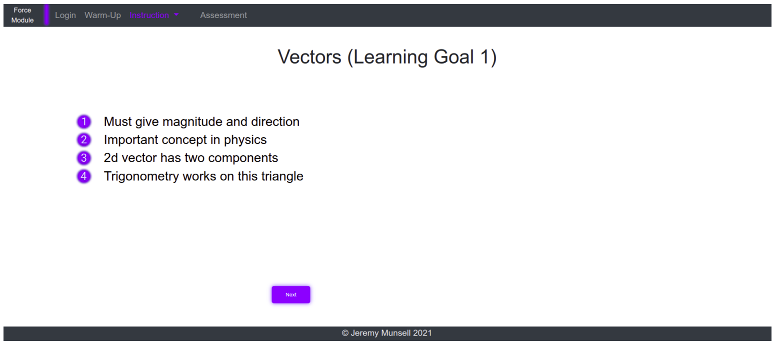
Since θ is the angle between $\vec{A}$ and the x-axis, A_x is the side adjacent to θ by definition. Likewise, A_y is the side opposite to θ by definition. Since A_y is opposite to θ,

$$sin(\theta) = \frac{A_y}{|\vec{A}|}$$

$$A_y = |\vec{A}|sin(\theta)$$

$$A_y = (5)sin(30^o)$$

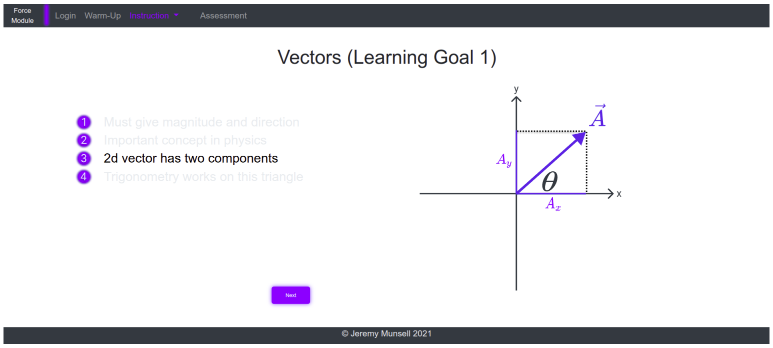
Figure A.4. Upon submitting the answer to the question, a video is played where the voice explains the logic behind the solution, and the mathematical steps of the solution are shown on screen in the HLG version (6.3.1).



Vectors (Learning Goal 1)

1. Must give magnitude and direction
2. Important concept in physics
3. 2d vector has two components
4. Trigonometry works on this triangle

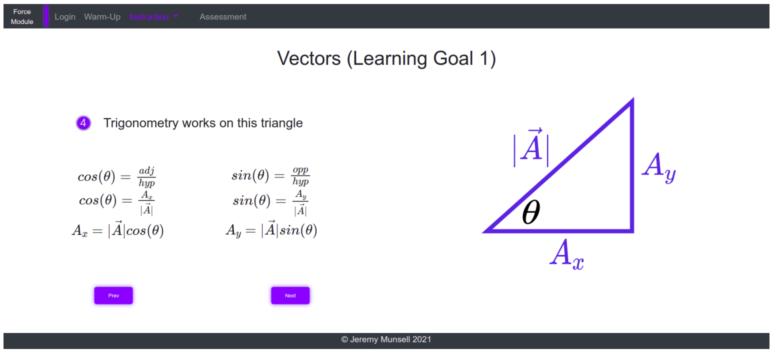
Figure A.5. The figure shows the state of the page before the student hovers their mouse over an item and after they move their mouse away from the item in the LLG version (6.3.1).



Vectors (Learning Goal 1)

1. Must give magnitude and direction
2. Important concept in physics
3. 2d vector has two components
4. Trigonometry works on this triangle

Figure A.6. Figure shows the user hovering mouse over one of the list items. The other list items are grayed out and a diagram is shown. When the student moves the mouse away from the list item, the page reverts to a state where just the list items are shown in the LLG version (6.3.1).



Vectors (Learning Goal 1)

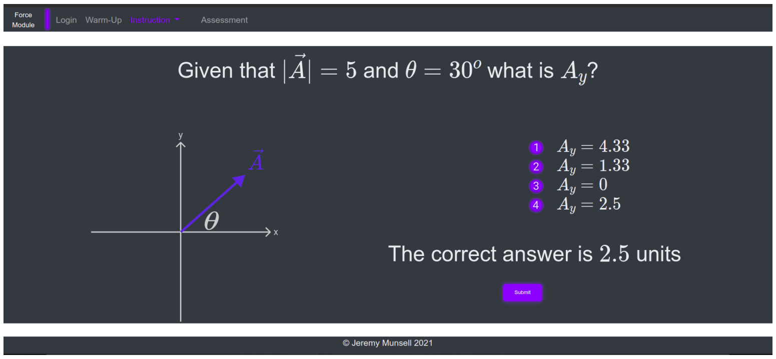
4. Trigonometry works on this triangle

$$cos(\theta) = \frac{adj}{hyp} \qquad sin(\theta) = \frac{opp}{hyp}$$

$$cos(\theta) = \frac{A_x}{|\vec{A}|} \qquad sin(\theta) = \frac{A_y}{|\vec{A}|}$$

$$A_x = |\vec{A}|cos(\theta) \qquad A_y = |\vec{A}|sin(\theta)$$

Figure A.7. Students are shown how to find the components of a vector in the LLG version (6.3.1).

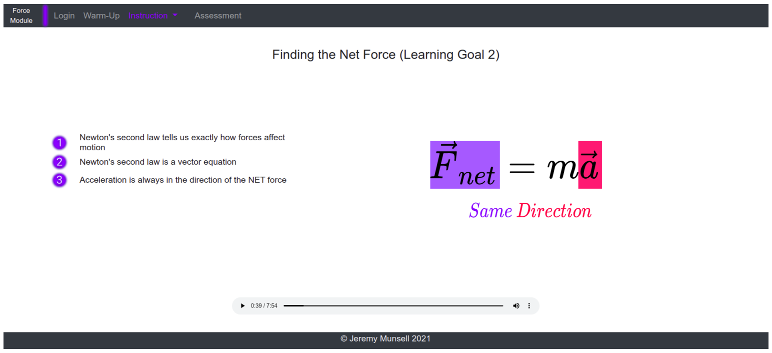


Given that $|\vec{A}| = 5$ and $\theta = 30^o$ what is A_y?

1. $A_y = 4.33$
2. $A_y = 1.33$
3. $A_y = 0$
4. $A_y = 2.5$

The correct answer is 2.5 units

Figure A.8. Students are shown product feedback only in the LLG version (6.3.1).



Finding the Net Force (Learning Goal 2)

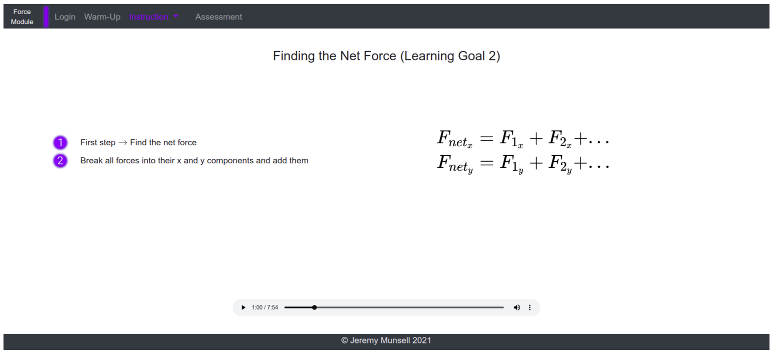
① Newton's second law tells us exactly how forces affect motion
② Newton's second law is a vector equation
③ Acceleration is always in the direction of the NET force

$$\vec{F}_{net} = m\vec{a}$$

Same Direction

© Jeremy Munsell 2021

Figure A.9. The figure shows a section of the module where Newton's second law is introduced in the HLG version (6.3.2).



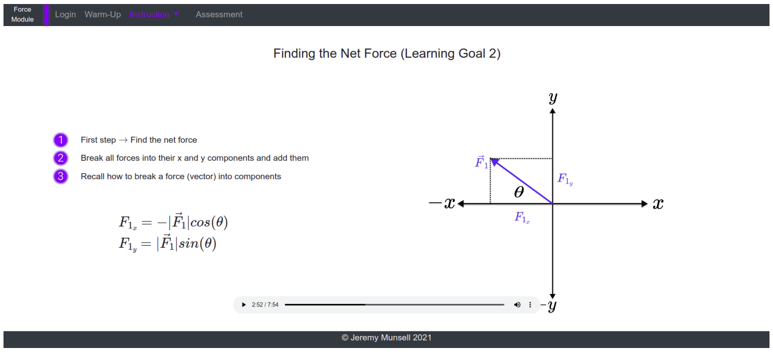
Finding the Net Force (Learning Goal 2)

① First step → Find the net force
② Break all forces into their x and y components and add them

$$F_{net_x} = F_{1_x} + F_{2_x} + \ldots$$
$$F_{net_y} = F_{1_y} + F_{2_y} + \ldots$$

© Jeremy Munsell 2021

Figure A.10. The definition of the components of the net force is shown in the HLG version (6.3.2).



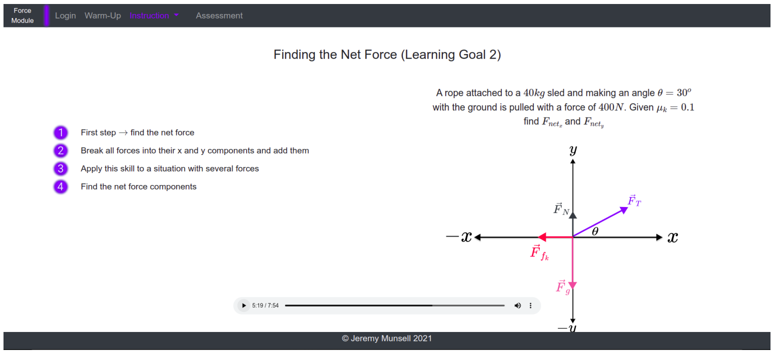
Finding the Net Force (Learning Goal 2)

① First step → Find the net force
② Break all forces into their x and y components and add them
③ Recall how to break a force (vector) into components

$$F_{1_x} = -|\vec{F}_1|cos(\theta)$$
$$F_{1_y} = |\vec{F}_1|sin(\theta)$$

© Jeremy Munsell 2021

Figure A.11. Vector decomposition is shown for vectors in other quadrants than quadrant 1, and with an angle defined with respect to both the x and y axes in the HLG version (6.3.2).

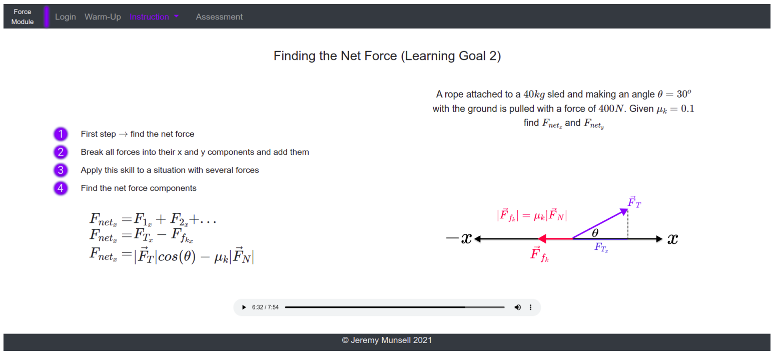


Finding the Net Force (Learning Goal 2)

A rope attached to a $40kg$ sled and making an angle $\theta = 30^o$ with the ground is pulled with a force of $400N$. Given $\mu_k = 0.1$ find F_{net_x} and F_{net_y}

1. First step → find the net force
2. Break all forces into their x and y components and add them
3. Apply this skill to a situation with several forces
4. Find the net force components

Figure A.12. Forces are added to an empty set of x-y axes one at a time as the voice in the video explains how the problem statement describes a force of that magnitude pointing in that direction until the force diagram is complete in the HLG version (6.3.2).



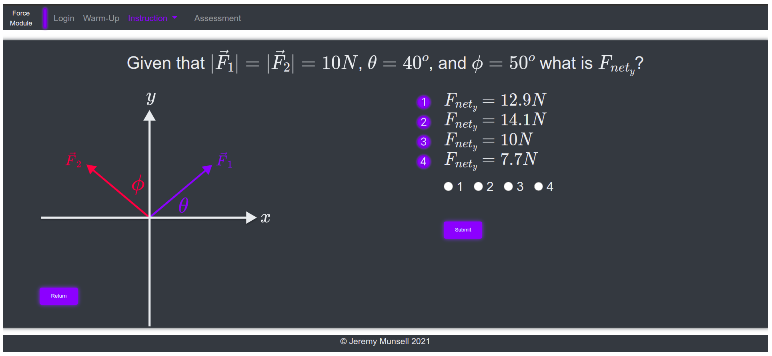
Finding the Net Force (Learning Goal 2)

A rope attached to a $40kg$ sled and making an angle $\theta = 30^o$ with the ground is pulled with a force of $400N$. Given $\mu_k = 0.1$ find F_{net_x} and F_{net_y}

1. First step → find the net force
2. Break all forces into their x and y components and add them
3. Apply this skill to a situation with several forces
4. Find the net force components

$$\begin{aligned}F_{net_x} &= F_{1_x} + F_{2_x} + \dots\\ F_{net_x} &= F_{T_x} - F_{f_{k_x}}\\ F_{net_x} &= |\vec{F}_T| cos(\theta) - \mu_k |\vec{F}_N|\end{aligned}$$

Figure A.13. The force diagram is used to write the x component of Newton's second law. The y axis and all of the forces without a x component are hidden, and each of the forces are described in the context of Newton's second law. This is repeated for y-axis and the y-component in the HLG version (6.3.2).

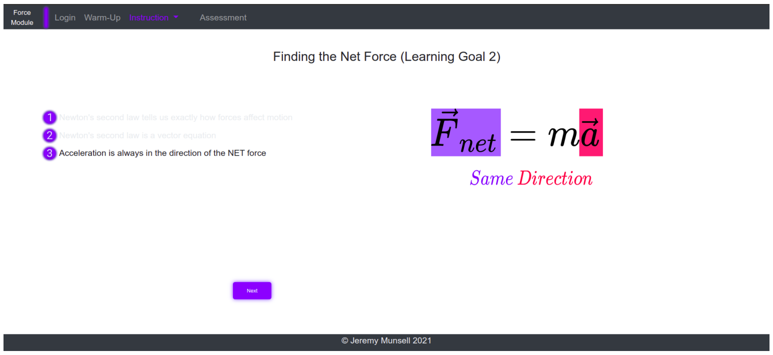


Given that $|\vec{F}_1| = |\vec{F}_2| = 10N$, $\theta = 40^o$, and $\phi = 50^o$ what is F_{net_y}?

1. $F_{net_y} = 12.9N$
2. $F_{net_y} = 14.1N$
3. $F_{net_y} = 10N$
4. $F_{net_y} = 7.7N$

● 1 ● 2 ● 3 ● 4

Figure A.14. The assessment question for the HLG/LLG net force sub-section (6.3.2).



Finding the Net Force (Learning Goal 2)

① Newton's second law tells us exactly how forces affect motion

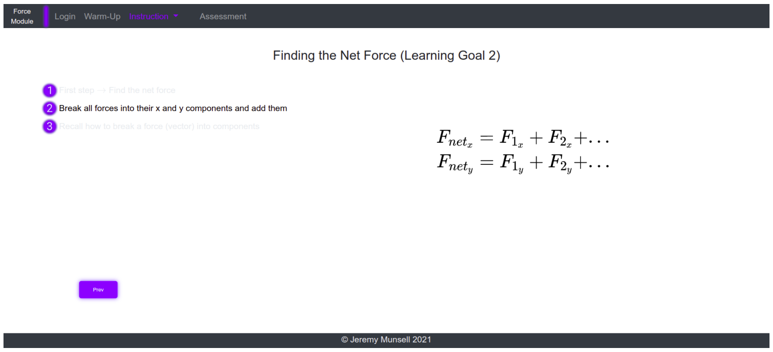
② Newton's second law is a vector equation

③ Acceleration is always in the direction of the NET force

$$\vec{F}_{net} = m\vec{a}$$

Same Direction

Figure A.15. The figure shows a section from the LLG version of the module where Newton's second law is introduced in the LLG version (6.3.2).



Finding the Net Force (Learning Goal 2)

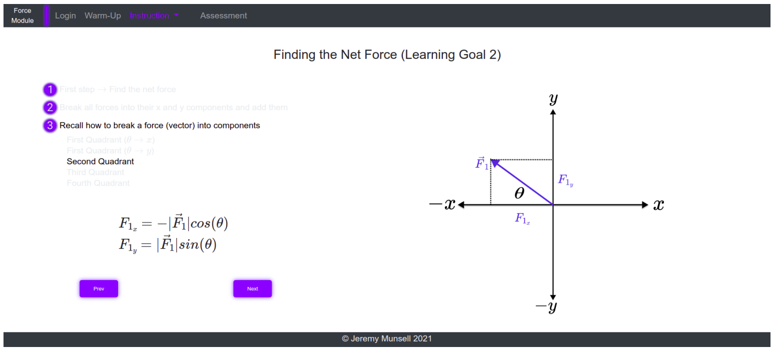
① First step → Find the net force

② Break all forces into their x and y components and add them

③ Recall how to break a force (vector) into components

$$F_{net_x} = F_{1_x} + F_{2_x} + \ldots$$
$$F_{net_y} = F_{1_y} + F_{2_y} + \ldots$$

Figure A.16. The components of the net force are defined in terms of the components of other forces acting in the problem (6.3.2).



Finding the Net Force (Learning Goal 2)

① First step → Find the net force

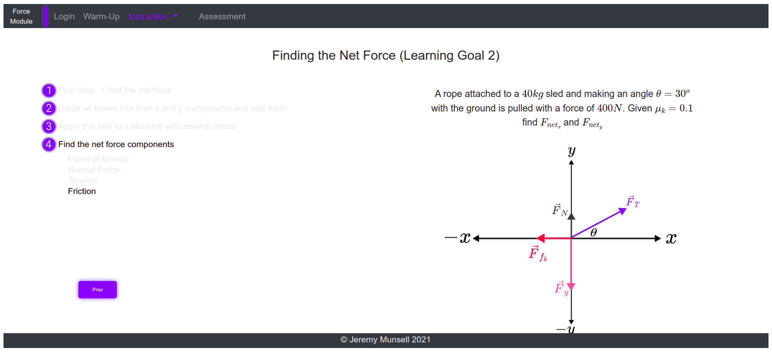
② Break all forces into their x and y components and add them

③ Recall how to break a force (vector) into components

First Quadrant ($\theta \rightarrow x$)
First Quadrant ($\theta \rightarrow y$)
Second Quadrant
Third Quadrant
Fourth Quadrant

$$F_{1_x} = -|\vec{F}_1|cos(\theta)$$
$$F_{1_y} = |\vec{F}_1|sin(\theta)$$

Figure A.17. The student hovers their mouse over an additional column of list items that pops down and they are allowed to see vector analysis for different combinations of quadrants and angles in the LLG version (6.3.2).

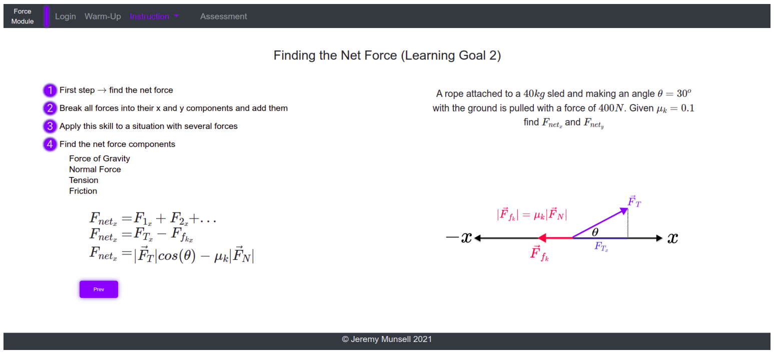


Finding the Net Force (Learning Goal 2)

① First step → find the net force
② Break all forces into their x and y components and add them
③ Apply this skill to a situation with several forces
④ Find the net force components
  Force of Gravity
  Normal Force
  Tension
  Friction

A rope attached to a $40kg$ sled and making an angle $\theta = 30^o$ with the ground is pulled with a force of $400N$. Given $\mu_k = 0.1$ find F_{net_x} and F_{net_y}

Figure A.18. As the student hovers their mouse over the list items in the pop-down menu, force vectors are added to the empty set of x-y axes in the LLG version (6.3.2).



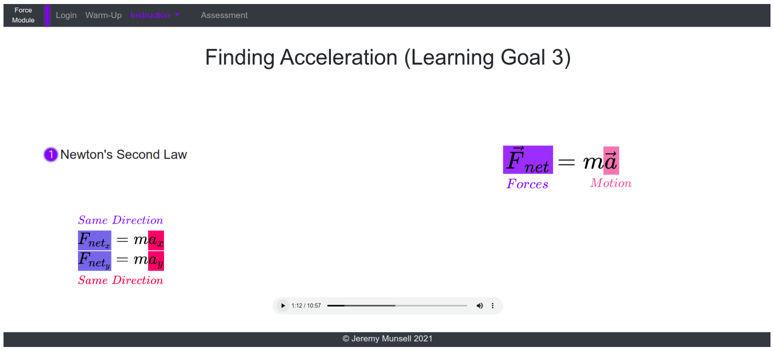
Finding the Net Force (Learning Goal 2)

① First step → find the net force
② Break all forces into their x and y components and add them
③ Apply this skill to a situation with several forces
④ Find the net force components
  Force of Gravity
  Normal Force
  Tension
  Friction

$$F_{net_x} = F_{1_x} + F_{2_x} + \dots$$
$$F_{net_x} = F_{T_x} - F_{f_{kx}}$$
$$F_{net_x} = |\vec{F}_T|cos(\theta) - \mu_k|\vec{F}_N|$$

A rope attached to a $40kg$ sled and making an angle $\theta = 30^o$ with the ground is pulled with a force of $400N$. Given $\mu_k = 0.1$ find F_{net_x} and F_{net_y}

Figure A.19. The figure shows what a student would see when they hover their mouse over the x-axis in the LLG version (6.3.2).



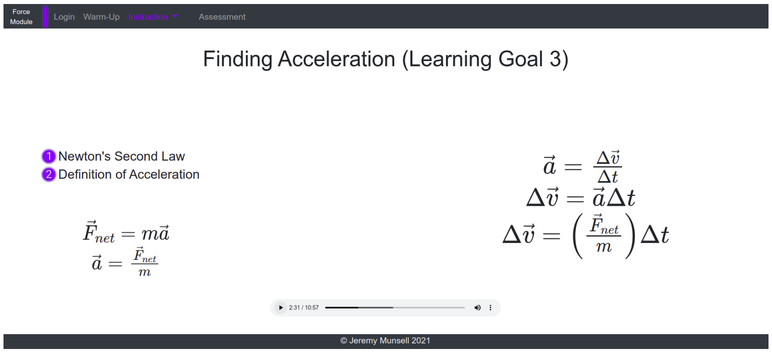
Finding Acceleration (Learning Goal 3)

① Newton's Second Law

$$\underbrace{\vec{F}_{net}}_{Forces} = m\underbrace{\vec{a}}_{Motion}$$

$$\overbrace{F_{net_x} = ma_x}^{Same\ Direction}$$
$$\underbrace{F_{net_y} = ma_y}_{Same\ Direction}$$

Figure A.20. The figure shows how Newton's second law being restated to emphasize that the the net force and acceleration are in the same direction, which also applies for Newton's second law in component form in the HLG version (6.3.3).



Finding Acceleration (Learning Goal 3)

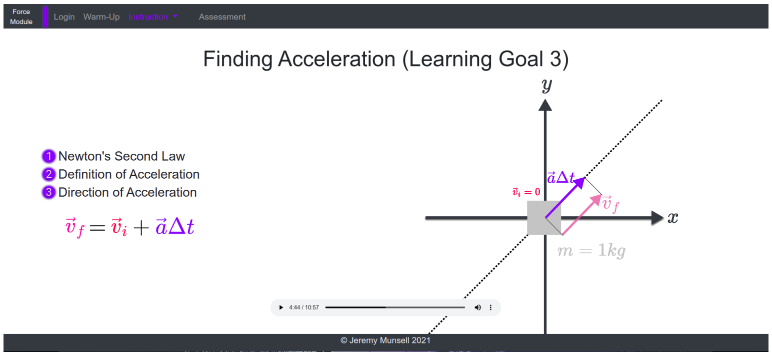
① Newton's Second Law
② Definition of Acceleration

$$\vec{F}_{net} = m\vec{a}$$
$$\vec{a} = \frac{\vec{F}_{net}}{m}$$

$$\vec{a} = \frac{\Delta\vec{v}}{\Delta t}$$
$$\Delta\vec{v} = \vec{a}\Delta t$$
$$\Delta\vec{v} = \left(\frac{\vec{F}_{net}}{m}\right)\Delta t$$

© Jeremy Munsell 2021

Figure A.21. The figure shows how the change in velocity of an object is related to the net force in the HLG version of the module (6.3.3).



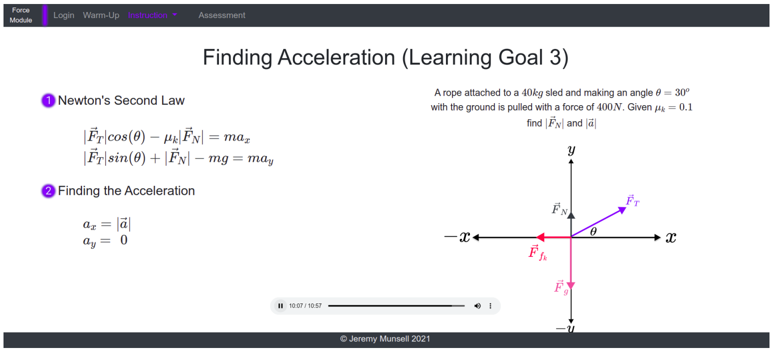
Finding Acceleration (Learning Goal 3)

① Newton's Second Law
② Definition of Acceleration
③ Direction of Acceleration

$$\vec{v}_f = \vec{v}_i + \vec{a}\Delta t$$

© Jeremy Munsell 2021

Figure A.22. The figure shows the final velocity being constucted given $v_\mathrm{i} = 0$ and the net force directed diagonally upward along the dotted line in the HLG version (6.3.3).



Finding Acceleration (Learning Goal 3)

① Newton's Second Law

$$|\vec{F}_T|cos(\theta) - \mu_k|\vec{F}_N| = ma_x$$
$$|\vec{F}_T|sin(\theta) + |\vec{F}_N| - mg = ma_y$$

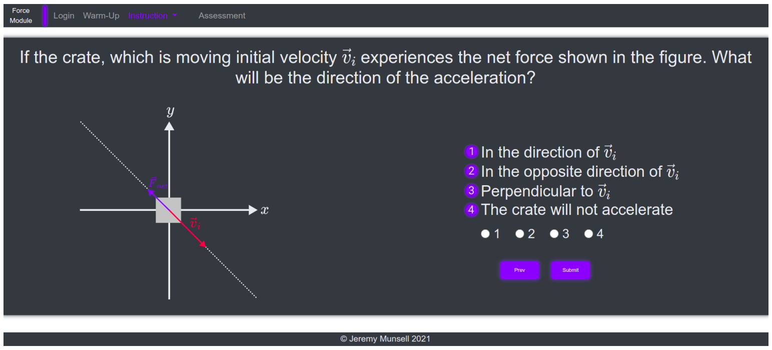
② Finding the Acceleration

$$a_x = |\vec{a}|$$
$$a_y = \ 0$$

A rope attached to a $40kg$ sled and making an angle $\theta = 30^o$ with the ground is pulled with a force of $400N$. Given $\mu_k = 0.1$ find $|\vec{F}_N|$ and $|\vec{a}|$

© Jeremy Munsell 2021

Figure A.23. The figure shows the direction of the acceleration being deduced from the problem statement in the HLG version (6.3.3).



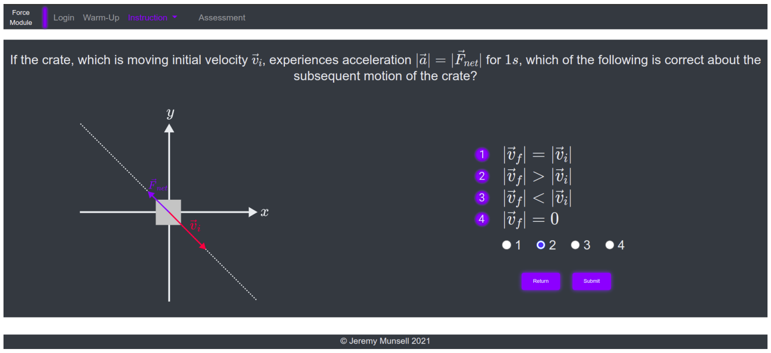
If the crate, which is moving initial velocity $\vec{v}_i$ experiences the net force shown in the figure. What will be the direction of the acceleration?

① In the direction of $\vec{v}_i$
② In the opposite direction of $\vec{v}_i$
③ Perpendicular to $\vec{v}_i$
④ The crate will not accelerate

● 1 ● 2 ● 3 ● 4

© Jeremy Munsell 2021

Figure A.24. The figure shows the first assessment question in the acceleration subsection of the HLG/LLG versions (6.3.3).



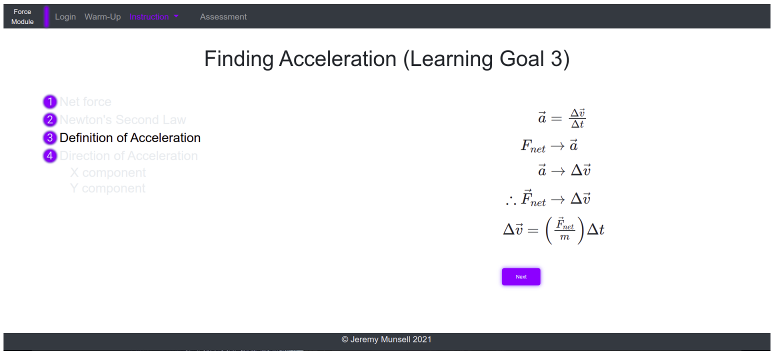
If the crate, which is moving initial velocity $\vec{v}_i$, experiences acceleration $|\vec{a}| = |\vec{F}_{net}|$ for $1s$, which of the following is correct about the subsequent motion of the crate?

① $|\vec{v}_f| = |\vec{v}_i|$
② $|\vec{v}_f| > |\vec{v}_i|$
③ $|\vec{v}_f| < |\vec{v}_i|$
④ $|\vec{v}_f| = 0$

● 1 ○ 2 ● 3 ● 4

© Jeremy Munsell 2021

Figure A.25. The figure shows the second assessment question in the acceleration subsection of the HLG/LLG versions (6.3.3).



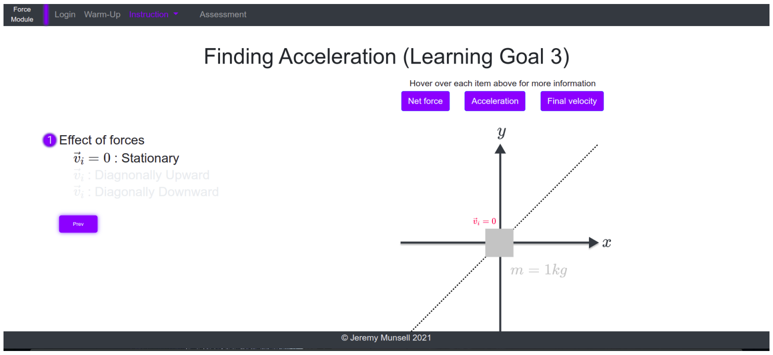
Finding Acceleration (Learning Goal 3)

① Net force
② Newton's Second Law
③ Definition of Acceleration
④ Direction of Acceleration
  X component
  Y component

$$\vec{a} = \frac{\Delta\vec{v}}{\Delta t}$$
$$F_{net} \rightarrow \vec{a}$$
$$\vec{a} \rightarrow \Delta\vec{v}$$
$$\therefore \vec{F}_{net} \rightarrow \Delta\vec{v}$$
$$\Delta\vec{v} = \left(\frac{\vec{F}_{net}}{m}\right)\Delta t$$

© Jeremy Munsell 2021

Figure A.26. The figure shows the change in velocity being related to the net force in the LLG version (6.3.3).



Finding Acceleration (Learning Goal 3)

Hover over each item above for more information

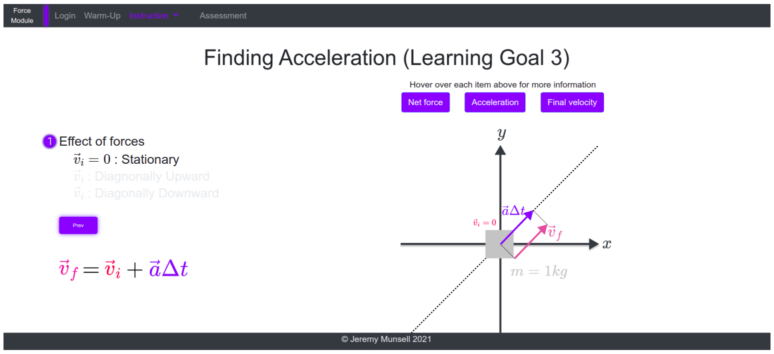
1 Effect of forces

$\vec{v}_i = 0$: Stationary

$\vec{v}_i$: Diagonally Upward

$\vec{v}_i$: Diagonally Downward

Figure A.27. The figure shows an interface where conceptual examples of finding the final velocity given initial velocity and the net force in the LLG version (6.3.3). This is the view when the student hovers their mouse over list items on the left highlighting a particular initial veloctiy



Finding Acceleration (Learning Goal 3)

Hover over each item above for more information

1 Effect of forces

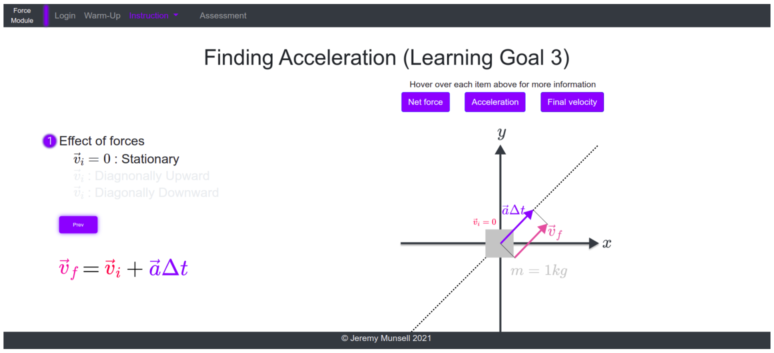
$\vec{v}_i = 0$: Stationary

$\vec{v}_i$: Diagonally Upward

$\vec{v}_i$: Diagonally Downward

$$\vec{v}_f = \vec{v}_i + \vec{a}\Delta t$$

Figure A.28. The figure shows interface where conceptual examples of finding the final velocity given initial velocity and the net force in the LLG version (6.3.3). This is the view when the student hovers their mouse over the purple buttons (top center).



Finding Acceleration (Learning Goal 3)

Hover over each item above for more information

1 Effect of forces

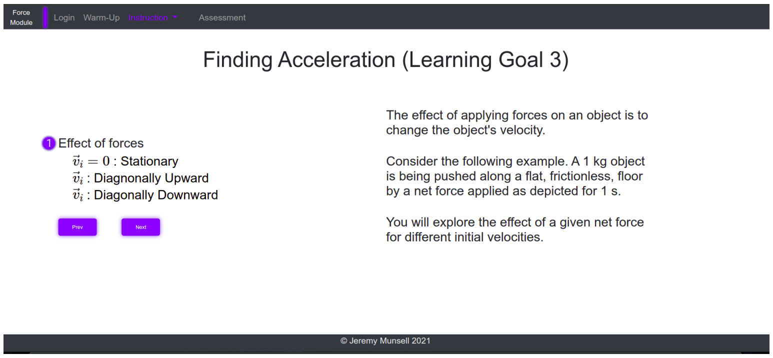
$\vec{v}_i = 0$: Stationary

$\vec{v}_i$: Diagonally Upward

$\vec{v}_i$: Diagonally Downward

$$\vec{v}_f = \vec{v}_i + \vec{a}\Delta t$$

Figure A.29. The figure shows the direction of the acceleration being deduced from the problem statement in the LLG version (6.3.3).



Finding Acceleration (Learning Goal 3)

① Effect of forces
$\vec{v}_i = 0$: Stationary
$\vec{v}_i$: Diagnonally Upward
$\vec{v}_i$: Diagonally Downward

The effect of applying forces on an object is to change the object's velocity.

Consider the following example. A 1 kg object is being pushed along a flat, frictionless, floor by a net force applied as depicted for 1 s.

You will explore the effect of a given net force for different initial velocities.

Figure A.30. The figure shows the default view of the conceptual example interface in the LLG version (6.3.3).