Predictive Analytics of Active Learning Based Education

by

Anuj Bheda

B.E., Nanyang Technological University (2011)

Submitted to the Integrated Design & Management Program
in partial fulfillment of the requirements for the degree of

Master of Science in Engineering and Management

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Signature redacted

Author ....

Integrated Design & Management Program
May 26, 2017

Signature redacted

Certified by ....

Dr. Abel Sanchez
Executive Director, MIT Geospatial Data Center
Thesis Supervisor

Signature redacted

Accepted by ....

Matthew Kressy
Director, MIT Integrated Design & Management Program

THIS PAGE INTENTIONALLY LEFT BLANK

Predictive Analytics of Active Learning Based Education

by

Anuj Bheda

Submitted to the Integrated Design & Management Program
on May 26, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering and Management

Abstract

Learning Analytics (LA) is defined as the collection, measurement, and analysis of data related to student performance such that the feedback from the analytical insights can be used to optimize student learning and improve student outcomes. Blended Learning (BL) is a teaching paradigm that involves a mix of face-to-face interactions in a classroom based setting along with instructional material distributed through an online medium. In this thesis, we explore the role of a blended learning model coupled with learning analytics in an introductory programming class for non-computer science students. We identify the features that were necessary for setting up the infrastructure of the course. These include discussions on preparing the course content materials and producing assignment exercises. We then talk about the various dynamics that were in play during the duration of the class by describing the interplay between watching video tutorials, listening to mini-lectures and performing active learning exercises that are backed by modern software development practices. Lastly, we spend time analyzing the data collected to create a predictive model that can measure student performance by defining the specifications of a machine learning algorithm along with many of its adjustable parameters. The system thus created will allow instructors to identify possible outliers in teaching efficacy, the feedback from which could then be used to tune course material for the betterment of student outcomes.

Thesis Supervisor: Dr. Abel Sanchez
Title: Executive Director, MIT Geospatial Data Center

Acknowledgments

I would like to thank my thesis advisor Dr. Abel Sanchez. Without Dr. Sanchez's efforts in putting his confidence into my ability to work on the thesis and support the development of the classroom infrastructure, this document would not have been possible. Dr. Sanchez is an exemplary individual who strikes out as a person with a deep passion for intellectual curiosity as well as a profound sense of human empathy.

I thank the entire course staff involved in the production of the 1.00/1.001 Engineering Computation and Data Science course; which forms the foundation of the case study documented in this thesis; for their unwavering patience and support in helping to carry out the functions required for the development of this experiment.

I must thank lecturer Matt Kressy and the entire course staff at the Integrated Management & Design Program, for their belief and faith me as an individual to be worthy of a coveted place in the inaugural edition of the program. Additionally, I wish to give a shout out to the cohort members of the program; a group of individuals who have been my classmates for the past two years and who individually represent a multitude of cultures, passions, and habits but collectively form a close-knit family.

Lastly, I would like to thank my friends, family, and foes. Without my experiences with them; which have ranged from extreme joy to sudden sadness; my journey in life would not have been possible.

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Problem Statement

What is an ideal blend of a learning model that is scalable as well as capable of delivering high-performance measures? These are the open-ended research questions that require extensive deliberation and debate. In this study, we take on the challenge of applying a blended learning model in a mixed level classroom setting focusing on the subject of introductory programming. The goal of the study involves using learning analytics to perform a continuous measurement of student performance throughout the duration of the course and be able to identify weaknesses in teaching methodology as well student comprehension. Based on the identification of such inflection points the next step would be to dynamically modify course content and adjust the difficulty of assignments to improve performance scores and achieve the desired level of high-quality education.

Why is it important and necessary to have such a learning model? It is without a doubt that education has helped lead the world in economic as well as social progress. However, a lot of our education systems are not built for the 21st century. The purpose of learning is to equip yourself with the skills necessary to function in a real world professional role [26]. However, a substantial number of these positions are being transferred or transformed either through globalization or automation. Thus, creating a learning model that is applicable for not just students at the undergraduate or graduate level but also at the adult education level is an urgent problem that needs to be solved with extreme rigor and urgency.

How do we come to a consensus on what the model should be? The area of research to find these alternate forms of educational models are well populated. Conferences, articles, journals and research papers around topics like web-based learning, active learning, massive open online classrooms, education data mining, etc are abundant. We start off with an exploration into blended learning, the theoretical framework for which is well documented. We also look into aspects of learning analytics that would help us frame our thoughts around building a predictive model based on the data collected. Our aim would be to pick out the best ideas from the traditional models and implement them in a real world setting starting with a course offered at the Massachusetts Institute of Technology. Through a series of quantitative and qualitative research of these ideas and implementations, the data collected and the insights gained would be a good predictor of whether such a model conforms to meet the needs of the future of education and its role in society.

1.2 Challenges in Education

Since the advent of university based education the traditional teacher led classroom model has been the norm. With a generation of students who have been exposed to the powerful forces of connectivity and computing technology; the traditional model can only seem to be regressive, formal and very direct. A survey on finding the core challenges that students face [10], identified cognitive understanding, becoming an active learner, and coping with reading material as the top three issues. Here the term cognitive challenge refers to the ability to grasp and understand content that is being taught. The issue around becoming active learners refers to the amount of effort students have to put in to participate in classroom discussions, and coping up with reading material refers to the challenge of narrowing down the material that is relevant for the course. If we take a look at the modern landscape of how learning is evolving, it is deemed to be more social, collaborative and passive [19]. Throughout the world of education, be it is K-12 schools or universities, a plethora of models are being tried and tested in order to discover the next baseline for classroom education. Personalized classrooms, massive open online classrooms, flipped classrooms, blended learning, etc are just a few examples of the methods being employed in schools and universities [18] [21] [17] [29]. The hope is that the challenges of 21st century education; be it with respect to the infrastructure of education, the delivery of education or the outcomes of education; are being rethought to cater to the needs of the current generation.

1.3 An Era of Innovation

The invention of the modern printing press by Johannes Gutenberg in 15th century Europe is considered to be one of the most prolific inventions in the history of modern human life. The advent of the printing press has led to an explosion in the spread of ideas and has brought about enormous economic and technological progress [8].

We are well into an era of a similar information and technological burst cycle. The advent of the internet and modern web technology has created an opportunity for an exponential growth in ideas, opportunities, and revolutions. Today, in the 21st century, as we look around us, industries ranging from Media to Finance to Healthcare to Transportation; are all being disrupted with innovative technologies, leaner business models and excellent user experiences [12]. Uber gives us access to get from point A to point B at the click of a button. 23andMe lets us view our entire genetic history using a simple DNA test. Robinhood allows us to trade in the stock market without being burdened by brokerage fees. Snapchat has become a visceral tool for the youth to engage with friends and family. Platforms like Facebook & Twitter have resulted in the proliferation of online communities that have spearheaded the task of building dynamic information pipelines.

Education is another such industry that has gone through multiple cycles of adopting disruptive technological models. According to CB Insights [1], in 2015 the ed tech industry witnessed a total of 511 deals amounting to a sum of $3.286 billion of capital invested. Though, like many other industries, a lot of those models have not performed as expected. However, the entire industry collectively has made leaps and bounds of progress when it comes to the accessibility, methodology, and effectiveness of imparting quality education.

1.4 Future of Learning

In Figure 1-1 we show a trend of ed tech companies on a spectrum of non-traditional to traditional methods with varying degrees of tech adoption.

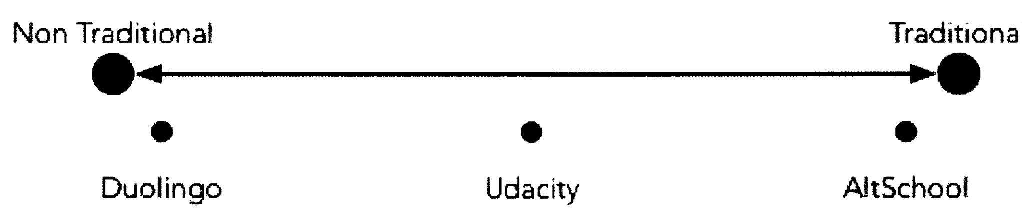


Figure 1-1: Spectrum of Educational Models

[1]Global Ed Tech Startup Deals And Funding See An Uptick - https://www.cbinsights.com/blog/global-ed-tech-deals-funding-q2-2016

AltSchool [2] is an educational institute with brick and mortar schools in California, New York, and Chicago. The schools focus on delivering a highly personalized learning experience for the students enrolled in their institute. Max Ventilla, who after having sold his startup Aardvark [3] to Google ended up as their Head of Personalization. After leaving Google, he started AltSchool with a mission to create a new elementary school system. They hire not only teachers but also computer programmers. These developers have created intellectual property in the form of a software platform that includes elements of management of school operations as well as pipelines to build personalized learning experiences. AltSchool represents a classic example of a startup sticking to traditional routes with a physical presence of interaction between teachers and students but with an added element of a software component to optimize the learning experiences of the students.

Udacity [4] is an another Bay Area ed tech startup founded by Sebastian Thurn, a senior executive at Google and a professor at Stanford University. Udacity started off with a focus on university styled lectures stemming from Stanford University's online course offerings, Today, it has turned into a powerful platform for learning vocational skills in the field of computer science. Their nanodegree programs offer classes in cutting-edge computer science research topics like deep learning, robotics and self-driving cars in addition to classes on learning mobile and web technologies. The firm offers degree certificates for successful participation in the classes and has partnered with large software companies to be a source of content ideas as well as a sink for many of their students who are looking for post training employment. Udacity serves as an example of how core academic topics are transforming into a

[2] AltSchool - https://www.altschool.com

[3] Aardvark (https://www.crunchbase.com/organization/aardvark) was a social search platform that allowed users to ask questions to their extended network of friends and family

[4] Udacity - https://www.udacity.com

form that is valuable for student learning as well as applicable with today's industrial needs.

Duolingo [5] is a mobile and web-based language learning tool. Luis von Ahn, then a professor at Carnegie Mellon University, along with a few of his graduate students started the company. They envisioned creating an application that served the purpose of allowing people to learn a foreign language while translating pieces of foreign language documents. The platform hosts more than a dozen languages now, with a total user base of 120 million who have collectively completed over 6 billion exercises. Their language certifications are applicable as valid test scores for admission requirements in multiple leading universities. Duolingo is an example of a company that has mastered the art of using a crowdsourced and gamified approach to making learning a new language simple, fun and efficient.

All the three models described above are great examples of disruptive and innovative technologies creating vital new experiences of learning and education. Traditional learning models comprising of instructor-led classroom lectures are being turned on their head to create room for more innovative methods in effective education. Education is no longer considered to be a one size fits all model, but is being highly personalized and curated for people with various needs, challenges, and difficulties.

1.5 Thesis Structure

The content that follows is a case study on the implementation of a blended learning model with learning analytics in a class on introductory programming principles. The thesis itself is broken up into six chapters.

[5] Duolingo - https://www.duolingo.com

Chapter 1 offers a glimpse into the world of education and how technology is helping transform the nature of learning and pedagogy.

Chapter 2 looks at the literature review in the subject with a focus on the aspects of blended learning and learning analytics.

Chapter 3 introduces the structure of the classroom and the components required to support the active learning methodology.

Chapter 4 talks about the infrastructure and data pipeline that was built to facilitate the publishing of assignments and data capture methods.

Chapter 5 delves deep into the analytical and predictive model created to help determine student performance.

Chapter 6 culminates the thesis with recommendations and potential work reserved for the future.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Review

2.1 Blended Learning

2.1.1 Definition

Since the advent of university education, teacher-led classroom based courses have been the norm. The classroom is reserved for a lecture delivered by the instructor through a presentation, talk or keynote. Any knowledge or wisdom that is expected to be imparted occurs during that time while the class is in session. Any discomfort or difficulty in understanding said knowledge or wisdom is not addressed immediately but is rather reserved for the students as a bridge they themselves need to cross. Given the state of affairs, such lectures have had a reputation for being inflexible as well as indifferent to the needs of individual students. With communication technology enabling classrooms to be held in a virtual environment, a new line of business in the form of distance learning education started proliferating in the higher education industry. The distance learning model offered on-demand learning, the convenience of location, cost efficiency and quickly gained prominence in the world of higher education. Over time, the merger of traditional classroom-based models and the technology afforded by innovations in the distance learning model resulted in the birth of the blended learning system, thus allowing educators to reap the benefits of both the models and mitigate their limitations [11].

A number of definitions have been put forward to determine what exactly is represented by the blended learning concept. Does it include a blend of instruction in various mediums? Does it afford personalized instruction? Does it completely flip the classroom model? For this text, we can safely assign the definition to be a learning system that includes elements of face-to-face interactions coupled with computer-mediated interactions [11]. As shown in 2-1, the blended learning model can be visualized as a combination of the traditional face-to-face interaction model accompanied by computer-mediated methods used in online or distance learning education.

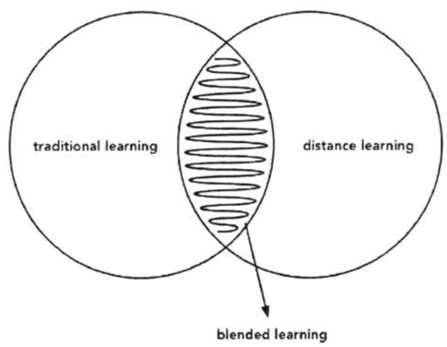


Figure 2-1: Blended Learning Intersection

2.1.2 Benefits

There are potentially many reasons why a blended learning model proves to be of more consequence in a classroom setting. The three most cited reasons are improved pedagogy, increased access and flexibility and improved cost effectiveness [11].

- Improved Pedagogy - A blended learning model allows a student to have multiple chances at reviewing, learning and practicing the material of the course. Individual learners come into a course at multiple levels of proficiency and have a varied speed of grasping the new content. A blended model with on-demand always available content allows individual learners to improve their understanding of the subject matter at a pace appropriate for them.

- Increased Access and Flexibility - Access to quality education remains an inherent problem across the world mostly due to location, timing, and infrastructure issues. A model that involves material being disseminated through an online medium helps alleviate the issues of time and location thus increasing flexibility for learners who would otherwise be tied down with constraints.

- Improved Cost Effectiveness - Saving on costs without harming educational quality is a prominent goal for any academic or corporate institution. Due to the nature of a mix of traditional classroom and distance learning methods, blended learning can be more cost effective since it does away with the overheads of physical infrastructure. The added cost required for building the digital infrastructure is more than offset by the cost savings.

2.1.3 Challenges

Blended learning as a concept started a few decades ago, but it is still under siege from traditional influences and has faced numerous challenging in setting the stone for the future of education. When it comes to designing the course content and develop the curriculum structure, a simple effort to clone classroom activities and put them in an online form have deemed to be not successful [14]. Blended learning gives instructors the opportunities to be more engaging, use alternative mediums and create a vibrant and lively experience. However, such experiences need to go through a formalized design process, where content and structure must be built with an understanding of the limitations of the traditional model. The process must follow a user-centric design process, where assumptions are established, research is conducted, concepts are generated, prototypes are built, feedback is retrieved, and the entire process is repeated till there is enough confidence that the design has achieved its intended purpose.

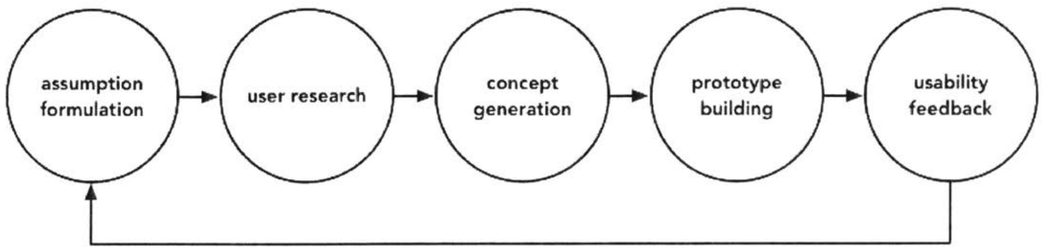


Figure 2-2: User Centric Design Process

A blended learning environment does not absolve the responsibilities of an instructor, on the flip-side, there might be more work that needs to be accomplished [14]. There has to be a right level of support coming from the instructor in handling both technical and non-technical issues. Care must be taken to ensure that the amount of live interaction and online interaction is set at an optimum level and is not biased towards any one side. The same should be made clear to the students so that they do not overcompensate for one part of the blended learning experience and completely avoid the other.

2.1.4 Applications

The use of blended learning models in the corporate world has been widespread. At IBM [16], the educators adopted a blended learning model for new manager training. Their philosophy around creating blended learning content revolved around a four tier learning model as shown in Figure 2-3. Tier 1 involved providing quick and easy access to all managerial material. Tier 2 comprises of an online learning module that immerses the managers in virtual simulations. Tier 3 brings together managers from across the world in an online forum to create a global collaborative learning space. Finally, Tier 4 is a face-to-face module that culminates the learning process by providing a space for deeper skill development. The learning model thus described was used in the preparation of the Basic Blue for Managers, a training program for newly minted first line managers. The learning process was broken into three phases. Phase 1 was a period of self-paced, online learning that lasted for 26 weeks. Phase 2 was reserved for a 5-day in-class session conducted throughout the company's global learning centers. Phase 3 was an online learning module similar to Phase 1 but was reserved for content that was more complex and specific to the skills and knowledge of the individual manager.

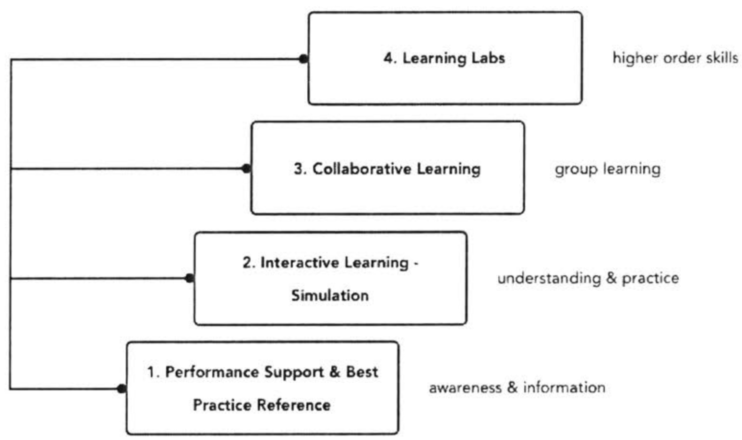


Figure 2-3: IBM's Tiered Learning Model

The evaluation of the model was an important point of their analysis, and they performed their measurement using the Kirkpatrick model on training impact Figure 2-4. The procedure consisted of rating the training model on five different levels. Level 1 corresponded to Reactions and was a measure of student satisfaction on course content and delivery. The student interviews revealed an unequivocal enthusiasm for the online and offline implementations of the program. Level 2 was around the aspects of Learning and based on the metrics they retrieved, over 96% of 600 participants achieved mastery in 15 knowledge-based tests. Level 3 related to Transfer and was intended to capture behavioral changes post a few months after course completion. Level 4 evaluated business impact and most participants perceived a sense of leadership improvement which in turn had a positive impact on business outcomes. Lastly, Level 5 was a measure of Return on Investment and captured the cost efficiency incurred by the organization due to the implementation of the program. The numbers estimated put down the degree of savings to the tune of more than a few hundred thousand dollars.

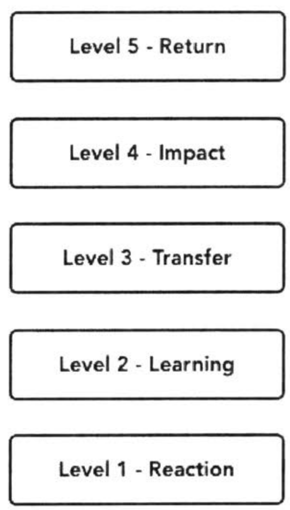


Figure 2-4: Kirkpatrick Training Evaluation Model

2.2 Learning Analytics

2.2.1 Definition

Business Intelligence software in the corporate world has allowed organizations to get a detailed insight into their operations, supply chain, revenue models, employee productivity amongst a host of other metrics. On a similar front, the field of learning analytics aims to be the empowering tool that utilizes big data methods to give institutions and individuals the power to augment student performance [7]. Learning Analytics is the described as the process of continuously collecting, measuring and analyzing student data to provide a valuable feedback loop back to optimize student learning and improve student outcomes [20]. Such a measurement effort can be easily construed with the advent of courses that are delivered through online mediums. Advances in modern software technology allow us to capture a complex number of specific user actions. Along with active research in the multitude of areas that led to the advent of learning analytics [5], the field is ripe for producing the effects that instructors have long awaited.

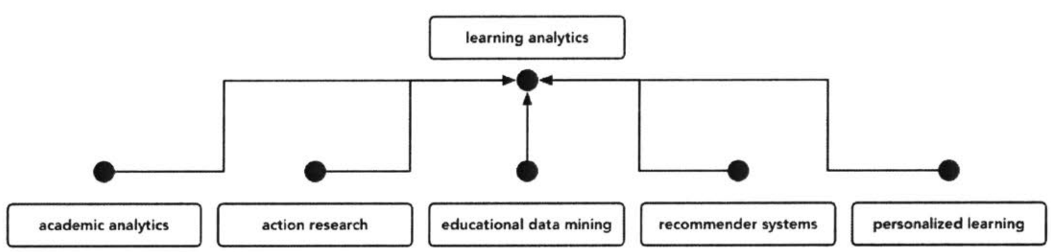


Figure 2-5: Learning Analytics Research Areas

- Academic Analytics - It refers to the gathering, storing and analyzing of data to help academic institutions make a decision regarding student performance, outcomes, retention, and personalization. The field leans towards more of a statistical evaluation of the data rather than creating a predictive capability.

- Action Research - Classroom instructors more often than not have an active interest in education research as a means to evaluate their work and provide insights for the future. The field of action research is more qualitative than quantitative and give instructors a tool to measure their performance.

- Education Data Mining - Extends the area of data mining to use large sets of educational data to find insights on ways to improve education quality, student outcomes, and teacher performance. The field makes use of the state of the art systems, methods and algorithms and is an active topic of interest for many.

- Recommender Systems - Such systems are widely used in a broad array of consumer applications. From watching movies to figuring out what to buy, recommender systems affect our daily decisions all the time. With rapid advances in machine learning algorithms, the field of recommendation systems will become stronger and more relevant in the near future.

- Personalized Learning - Individuals come with a varying degree of pace at which they learn. The goal of personalized learning is to modify the course content such that it caters to the learning capabilities of individual learners.

2.2.2 Method

The method for performing the task of building learning analytics can be simplified into a three step process as shown in Figure 2-6 [5]. The initial step is to capture and collect data stemming from the actions performed within the context of course related activities. The data would be the foundation on which future analytical insights would be obtained. Such data points could be captured through a learning management system, online learning tools and even obtained through open data sets.

A pre-processing step allows the system to separate the signal from the noise and get rid of redundant or non-useful data points. Using the pre-processed data as a starting point, the process of analysis would involve applying data mining, statistical inference and machine learning techniques to get insights, patterns, and discoveries out of the data. A few possible methods that could be used are regression, decision trees, k-means, support vector machine or neural networks. The obtained information is then synthesized using visualizations or recommendations to chart out an actionable map of steps. These steps would change the dynamics of the course to move certain measurable metrics up or down. A post-processing step clears the ground for iteratively performing the steps in the cycle ensuring that the feedback loop is constant and continuous.

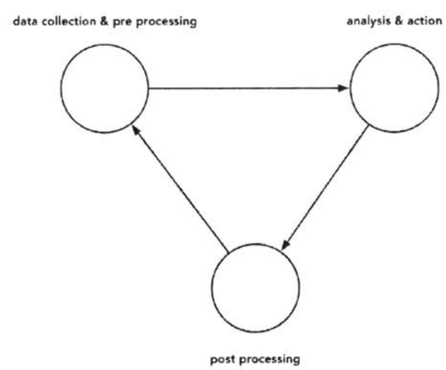


Figure 2-6: Learning Analytics Method

2.2.3 Benefits

The primary benefit of the implications of applying the feedback obtained through learning analytics would be to measure, comprehend and improve student performance. The improvement need not be measured on a simple quantitative level but could also be extended to include qualitative factors. The factors would collectively help to influence the learning experience of the students. If the feedback is given back to the students in a curated manner, the curriculum could be personalized such that each individual is given a unique blend of coursework to bridge the gaps in perceived knowledge.

On a secondary level, multiple benefits have been determined to prove consequential. Foremost, understanding what is working and what is not would improve the course curriculum not just for the class in session but also for future generations. The feedback obtained would be tremendously consequential in helping evolve the curriculum such that it meets the needs of the students. Apart from this, instructors will be able to get a glimpse into their strengths and weaknesses and adjust their teaching styles to match with the upward measures of student performance.

2.2.4 Challenges

Without a doubt, the field of learning analytics is growing and will occupy an important place when it comes to education. With growth comes pain and some of the challenges revolve around keeping a good leash on the hard part around data science as well the soft part around user focus [9].

- Leaning Pedagogy - It would be imperative to understand how students learn and this would require a focus on understanding the methodologies which are effective and ineffective when it comes to student learning. The value of the insights gained after the data analysis would have to be applied under a practical context of keeping in tune with the learning sciences.

- Relevant Datasets - As the field advances and makes breakthrough discoveries it would be pertinent to start adopting a set of data points that includes not only static information generated but also go further down the stream to understand the context in which the students were learning. The merging of such data points would help improve the richness of the analysis.

- Learner Focus - The analysis performed should be done from the perspective of the student learner. What might be good to increase performance scores, might be disadvantageous for the students. Hence, any metric evaluation should be done keeping in mind the holistic growth of the students.

- Ethical Issues - Data privacy is a big issue in almost all industries that utilize big data. The issue becomes even more sensitive in the context of an education environment since it is important that student privacy is protected. Effective policies must be implemented to stop the potential misuse of data.

2.2.5 Applications

A successful implementation of integrating learning analytics to deliver improved student success was in the form of Course Signals developed at Purdue University [1]. The aim of the project was to identify weak performers and intervene in early to prevent them from losing focus and dropping out. The authors devised the program as a method to ingest data from the university's learning management system, prior academic performance, and individual student characteristics leading to the development of a predictive model. The model utilized what they called the student success algorithm, and it was capable of classifying students into three buckets. Each bucket indicated the risk of the student being unable to participate and complete the course successfully. Feedback to the student was relayed back in the form of notifications on the learning platform, emails, text messages and even face-to-face meetings with the instructor. The results of the program indicate that they were successfully able to see an increase in the percentage of grades that were A and B and a reduction in grades C, D & F. The feedback from the instructors was highly positive and the program has been extended to a large number of courses conducted on campus. The feedback from the students was measurably positive because it genuinely improved their performance.

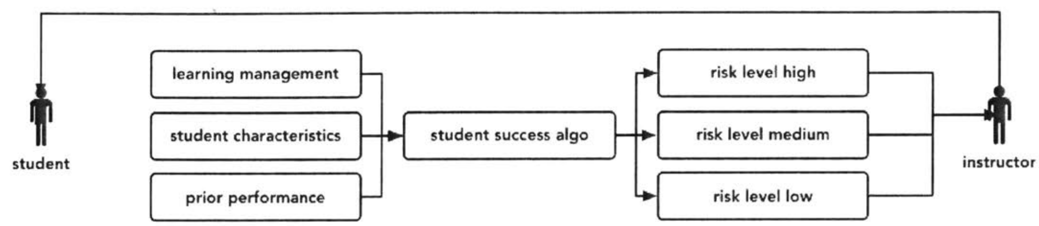


Figure 2-7: Course Signals at Purdue

In a study on measuring student behaviors in open-ended programming tasks [4], the author employed the use of a programming environment that allowed actions such as key presses, button clicks, code compilations to be recorded into a log file. The participants were undergraduate sophomore students, and the programming assignment included a task to model a scientific phenomenon of their choice. At the end of the assignment period, the individual log files from each of the participants were collected and analyzed. All in all, there were more than 9 million events captured, out of which a predominantly high number of them were non-code events. Based on the filtered code only events, the author was able to recreate a frame-by-frame picture of the student's progression as they made their way through the assignment.

Based on the model that was devised, the author was able to determine the unique strategies used by the participants 2-8. The seven strategies they found were stripping down code to a bare minimum; starting with a template and adding procedures; periods of inactivity during which they were either browsing other solutions or thinking about the solutions; linear growth indicating a trial and error method to come to the solution; sudden jump in code size when they copied a piece of code and finally a cool down period where formatting, indentation, and variable naming were taken care of. The author suggests that based on these coding strategies, one might be able to devise a set of support systems to help each individual profiles. Participants exhibiting the expert profile might be more inclined on getting easy access to advanced documentation while amateurs might be more inclined on getting additional sample code snippets.

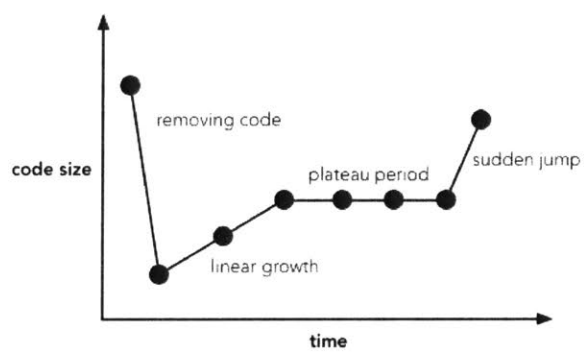


Figure 2-8: Coding Strategy

In a similar study that involved capturing log data and determining how changes in programming activity relate to the learning process and quality of code [3]; the authors use a visual programming environment to get the participants to build a soccer playing robot. The participants were high school students, and they worked in teams to accomplish the task. Every time the program underwent a change, the program state was captured along with its time stamp and user identifiers. The captured program states were then clustered into a couple of categories using measures of actions, logic, coverage, length and quality. Each category was given a distinctive tag such as minimal, logical, compact, balanced, etc. Each such tag corresponded to the average value of the number of features seen in the category. These categories were then mapped out based on their flow from one category to another, and their transition states identified the most likely path participants would take once they were in a particular state Figure 2-9.

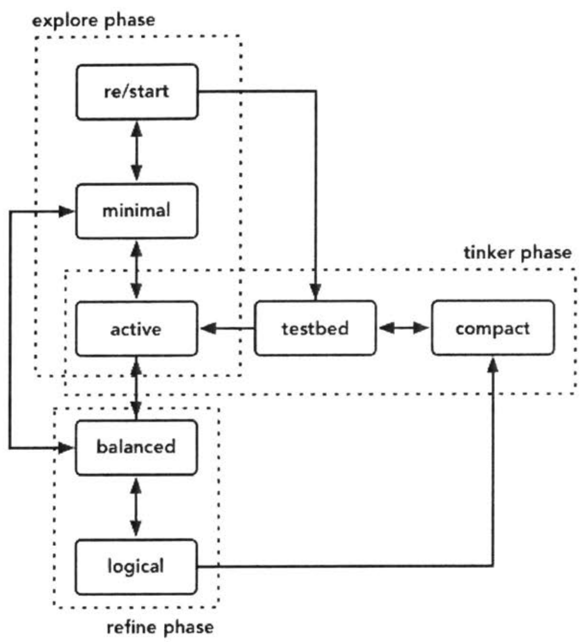


Figure 2-9: Program State Category Transitions

The students begin in the start category. Adding a few primitive logical functions moves them to the minimal category. From the minimal state, they would either transition to the balanced category or the active category. In addition to the analysis on tracking paths, the authors then identified regions where students are engaged in either exploring, tinkering or refining their program to reach the goal of creating a functional bot. Using these two dimensions of inferences, the authors were able to answer their research questions.

Following the code snapshot analysis paradigm, another research study involved using a scoring function for each code snapshot and applying linear regression on the scores as a prediction model to calculate the future performance of the students [28]. The scoring function took in a series of code snapshot pairs, compared their changes and returned an average score. The code compilations were snapshots of file changes that were pre-processed to remove snapshots that were identical or were the result of comment or deletion fixes. The scores were in the range of 0-1, with 0 indicating 0 errors during compilation and signified the hallmarks of a strong programmer. Increases in the score value would signify and highlight a weaker programmer.

Continuing the trend of capturing code snapshots, a group from Stanford University [27] captured program submissions made for the Hour of Code exercises on code.org and analyzed the submission data to predict future performance. They employed the principles of representation learning and knowledge tracing. Representation learning refers to the idea of using raw input data for machine learning processes instead of relying on human expertise to extract features out of the input data. Knowledge tracing refers to the concept of predicting student performance at a time t_n based on data available from time t_0, t_1, t_2 ... t_{n-1}. The dataset consisted of more than 1 million code submissions. The code submissions were converted into their Abstract Syntax Tree representations Figure 2-10 and used as inputs for the model. The model made use of a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). The output of the model was a binary classification classifying the success of the student's ability to solve the next exercise.

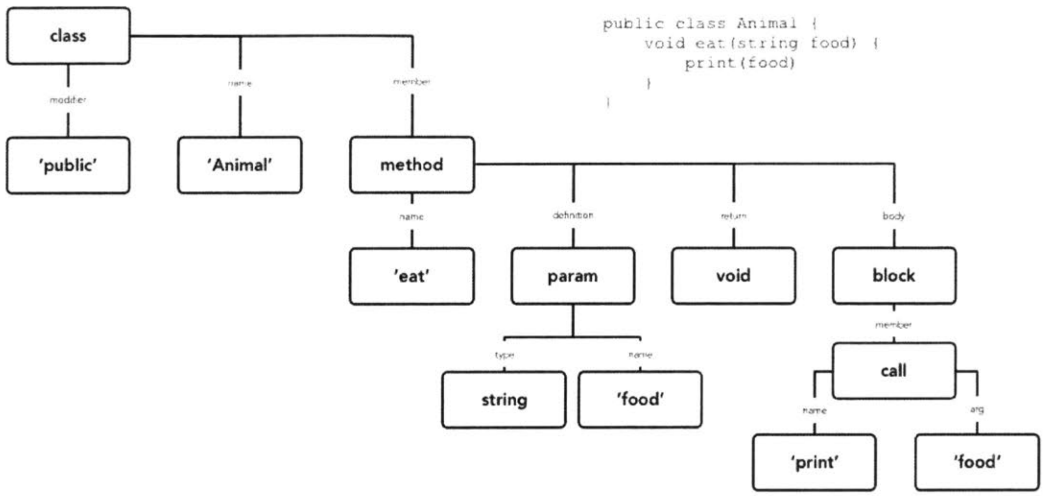


Figure 2-10: Abstract Syntax Tree

Another interesting study was done in the form of predicting student grades using freestyle comments data [23]. The authors devised a method to collect comments data on each lesson conducted during the course. Using semantic analysis on the comments data, the authors were able to create an input matrix of four dimensions that captured the user sentiment of the comment for that particular lesson. The input matrix was then applied to an artificial neural network where the output was in the form of a bucket of grades Figure 2-11. The model provided an accuracy of approximately 80% which was higher than their previously measured accuracy of 66% based on a k-means clustering model.

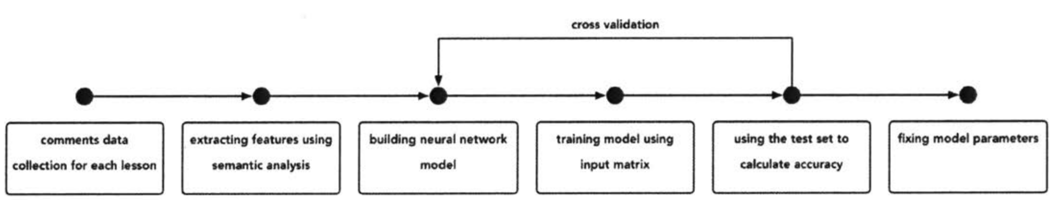


Figure 2-11: Student Grade's using Comments Data

Chapter 3

Classroom Dynamics

3.1 Course Description

1.00/1.001 was an undergraduate + graduate level course on engineering computation and data science. The course intended to build strong fundamentals of working with basic computer programming syntax, introduce the concepts of data science/machine learning and work through the world of web platforms/libraries that have become the backbone of the modern internet. The aim of the class was to cast a wide net and cover a broad array of topics that are essential for building up the necessary expertise to create functional applications. The philosophy of the course at the time was to build academic rigor through practical applications rather than theoretical expertise. Such rigor was built through a blended learning model comprising of preparatory videos, classroom mini-lectures, active learning exercises, homeworks and a final project.

The relevancy of this course could not be discussed at a better time. The world is moving to a stage where repetitive, mundane tasks will be automated to a degree even further than what we have witnessed so far. In the world of manufacturing, the industry has been upended through machinery and mechanized equipment. With the recent developments in artificial intelligence, computer vision and natural language processing, jobs ranging from truck drivers to store assistants are ripe for being taken over by automated machines. Underlying these disruptions is a core skill that revolves around the field of computer science and engineering. The Bureau of Labor Statistics estimates by 2020 over 14 million computer-related jobs will have openings only to be supplied by a pool of 400,000 graduates [1].

Focusing on that need to build up computational expertise, the course is an introduction to the world computers and its application. The student population of the course comprises mostly of non-computer science students who otherwise would not get a chance to work their way through programming assignments. The hope is that students appreciate and understand the material discussed through the topics covered in class and in conjunction with their core disciplines of civil engineering, material sciences, architecture or management, are able to meet the needs and challenges of an ever changing world that is becoming more and more dependent on computers and technology.

3.2 Medium of Instruction

3.2.1 History

Historically, the subject has been taught in C++ as well as Java, but a revamped version of the course launched last year changed the medium of instruction to JavaScript. JavaScript as a programming language has turned out to be the fundamental building block of the modern web. The language was first introduced in the Netscape Navigator web browser in the early 1990's and has moved up to version 6 today. Support for the language is well established on the browser front for both desktop and mobile computing environments. Leading universities around the world are adopting the

[1] Computer Science is for Everyone! - https://obamawhitehouse.archives.gov/blog/2013/12/11/computer-science-everyone

language due to its ubiquitousness and strong community [2].

3.2.2 V8 Engine

The success of Google Chrome as a web browser resulted in an offshoot project called Node. The Node architecture, as shown in Figure 3-1, is essentially a virtual environment that is powered by the V8 engine, the same engine that lies underneath Chrome. Node has allowed the execution of JavaScript as a standalone application and is today widely used as a backend server component. Thus, applications written in JavaScript not only have support on the client side frontend through web browsers but also on the backend server side through Node.

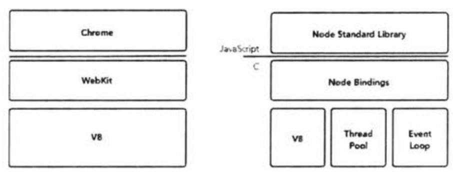


Figure 3-1: V8 Offshoots

3.2.3 Full-stack

Despite some of the drawbacks with respective to type management, leniency in syntax and different implementations of the engines running JavaScript [15]; the portability that the language offers, a strong community of developers building libraries along with its full-stack [3] nature for frontend and backend development; the course exclusively focuses on building the foundational experience in the syntax of the language as well as familiarity with orchestrating a multitude of platforms and libraries

[2]Stanford University being the recent institution to adopt the language as part of its introductory programming class - http://www.stanforddaily.com/2017/02/28/cs-department-updates-introductory-courses/

[3]The term full-stack development largely describes the phenomenon of engineering web applications with a client facing user interface and a server side implementation

in order to allow individuals to build high fidelity prototypes and even full-fledged web applications.

3.3 The Role of DevOps

3.3.1 Waterfall

Software development practices can be broken up into five stages as shown in Figure 3-2. Starting with the initial design phase, software engineers get a good grasp of the specifications of the product that is to be built. In the development phase, engineers start the work of writing production code to get the system up and running. The code is then put through a rigorous testing suite to catch any critical bugs after which the system is deployed for production usage. The production system is then continuously monitored for spikes in usage, anomalies in performance metrics and to keep a general check on the health of the system. Such a process has been termed as the waterfall software development process. This is how production software development started in the early 80's, and it provided a good structure, shape, and stability to the development lifecycle in the early days.

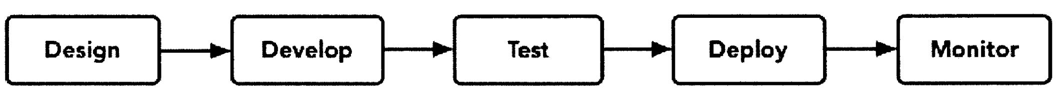


Figure 3-2: Waterfall Software Development

3.3.2 Agile

Unfortunately, the waterfall process was rigid. It was difficult to make changes further along in the lifecycle process since it was an expensive affair when a particular feature had to be redesigned or a significant use case failed to pass a test case. To fix these gaps, a new paradigm emerged in the software engineering community. The agile manifesto [22] started off as a set of values that encouraged minimalism in processes, execution over documentation, collaboration over negotiation and being accepting and responsive to change. The values introduced processes that led to a better definition of user design problems, greater collaboration between developers leading to rapid iteration along with continuous testing processes in place Figure 3-3. All in all, it resulted in shorter development cycles and was deemed to be a more effective strategy to build software products.

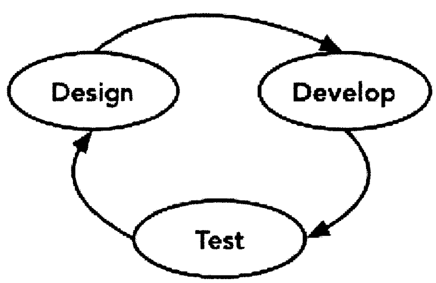


Figure 3-3: Agile Software Development

Extending the idea of continuous iteration and constant testing, these processes were applied to the deployment phase of the lifecycle process. This is what has led to the concept of DevOps, which essentially refers to the idea of agile methods implemented during the deployment phase of the software development process. A good DevOps routine would be to use version control to manage software development, engage in test driven development such that every line of code is well covered, continuously pushing the compiled build through an automated testing process and using containers to perform the task of deploying the application on a cluster of servers.

We use various ideas from the DevOps community to help architect the infrastructure of the course. The main goals of the exercise are to create an easy to use medium for students to work on their assignment and create a scalable data capture method such the feedback loop can be well established.

3.4 Automated Grading

3.4.1 Open Judge

The core idea of automated grading stems from the concept of online judges used in the world of competitive programming [6]. Online judges accept a piece of code submitted by the programmer and perform a series of tests to check for the correctness of the code as well the scalability of the code. It is as important for the submitted code to return the correct answer, as it is for the code to give the answer in a minimum amount of time. More recently the concept of online judges has been extended to web applications helping programmers prepare for software engineering interviews [4] and is also used on online education platforms that teach the fundamentals of programming [5]. Taking a similar stand, we rely on the integration of three independent tools to achieve the dual purposes of assignment efficiency and automated grading.

3.4.2 Version Control

Version Control (VC) is the concept of keeping a record of any changes in code that take place over the duration of the project's development lifecycle. Version control helps ease the process of collaboration when you have a multitude of engineers working on a variety of software features, the code for which all resides in a single repository.

[4] HackerRank is an example of an application engaging in this practice

[5] Codecademy is an online education platform which makes use of an online judge to help students learn

We make use of GitHub for the purposes of version control in the class. Each of assignments is modeled in the form of a single repository. The students act as external contributors and are able to get a copy of the assignment code as well as submit their solutions back to the platform. Detailed specifications on its usage will be discussed in the next chapter.

3.4.3 Test Driven Development

Test Driven Development (TDD) is the idea of building software in incremental steps and ensuring that each such incremental step is well accounted for using a series of unit tests. TDD ensures that the new code which is written does not include breaking changes when published. It is now considered to be a widely well-rated standard for good software development practices.

We make use of Mocha and Chai for building a testing environment in each of the assignments. Mocha is a tool that allows tests to be written in the behavior driven style. Chai is an assertion tool that checks for the validity of the output type to match the expected type.

3.4.4 Continuous Integration

Continuous Integration (CI) is the concept of strictly following the principle to commit and push code in incremental batches of working functionality such that it passes a suite of previously defined tests. This ensures that the system is built using small incremental batches of working code and helps avoid unforeseen circumstances related to large pushes resulting in massive failure.

We make use of Travis-CI for our integration needs. As mentioned in the previous section, students use GitHub to make a submission on their assignment deliverables.

Every single contribution that a student makes is made to go through the Travis-CI integration process to check for validity of the code by making sure that the code passes a set of predefined tests.

3.4.5 Data Pipeline

Results from the output of the integration tests are stored in a custom made web app with an underlying data store. The results collected serve as the basis on which we can measure student performance, and its detailed usage will be discussed in a proceeding chapter.

We use a multitude of tools and platforms to build the data pipeline infrastructure. On the frontend, we make use of Jade, Stylus, and Angular to render the appropriate user interface. On the backend side, we make use of Express along with MongoDB to create the routing and data layer for the web application. Gulp is used for build automation, and the entire application is hosted on a medium sized instance on Google Cloud Compute. Additionally, even though we had the choice of creating a containerized environment using Docker, we use a simple production and development build identified through environment variables and managed using PM2.

3.5 Curriculum Structure

The course is worth 12 units of engineering credits and lasts the entirety of a full spring semester. The overall structure can be broken down into five major components as shown in Figure 3-4. Prior video snippets, classroom meetings consisting of mini-lectures and active learning exercises, weekly homeworks, two-term quizzes and a final project.

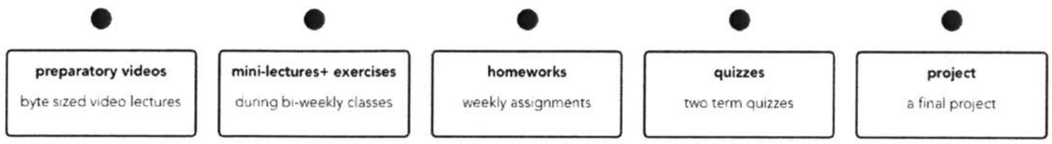


Figure 3-4: Curriculum Structure

Each classroom meeting is seeded with preparatory materials in the form of online videos. The classroom meetings happen twice a week with class participation required for the active learning exercises. The weekly homework assignments focus on a critical part of the content discussed during the classroom meetings. The quizzes include material that tests the students on material discussed in the two halves of the course and finally, the project is a two to three member group effort where the students are encouraged to pick up an open-ended problem statement and build a fully functional application tackling the problem.

3.5.1 Preparatory Videos

These are byte sized video lectures aimed at introducing the concepts to help the students build familiarity with the topic of discussion. The videos can also be used as a source of revision material post classroom discussion. These videos are used as a tool to equip students with the knowledge required to bridge the gap of level at which the classroom discussions and active learning exercises take place.

3.5.2 Mini-Lectures & Active Learning Exercises

Table 3.1 presents the various topics covered in class along with their respective tags. The course starts off with a basic introduction to the world of computing along with going through the setup instructions of the developer environment. The next phase involves going through basic programming *fundamentals* of working with arrays, objects, and functions. Next up are exercises on manipulating *data* that are available in the form of objects and arrays and using the same to build bar charts and scatter plots using an external charting library. We then talk about patterns involved in *software construction* and focus on callbacks, asynchronous computation, and package management. Next up are exercises on *machine learning* topics like k-nearest neighbors, k-means, and linear regression. To culminate the software portion of the topics we end with a discussion on databases, rest API's and how one can combine the various packages and libraries to build an online *web* service. Lastly, the class introduces the notion of *internet of things* and explains how hardware devices can work well in tandem with software components.

Table 3.1: Lecture Schedule

Lecture	Topic	Tag
Lecture 1	Course Overview	intro
Lecture 2	Arrays & Objects	fundamentals
Lecture 3	Functions	fundamentals
Lecture 4	Data	data science
Lecture 5	Visualization	data science
Lecture 6	Callbacks	software architecture
Lecture 7	Async Computation	software architecture
Lecture 8	Package Management	software architecture
Lecture 9	kNN & kMeans	machine learning
Lecture 10	Naive Bayes & Linear Regression	machine learning
Lecture 11	Databases	web
Lecture 12	REST	web
Lecture 13	Service Orchestration	web
Lecture 14	Sensors & Devices 1	iot
Lecture 15	Sensors & Devices 2	iot

Each lecture was split into a batch of mini-lecture and corresponding active learning exercises. Out of the 90 minutes scheduled for each lecture, you could have three mini-lectures lasting for around 10 minutes, each followed by an active learning exercise lasting for 20 minutes. A tentative split of time is shown in Figure 3-5.

The exercises involve starting off from a starter code template which includes instructions to write the functions required to get the tests to pass or to visualize a certain type of output. The validity of the functions could be checked using a locally available test suite. Each of the active learning exercises was distributed and collected back using GitHub. Grading was based on the test suite provided as part of each exercise and its detailed implementation will be discussed in a later chapter.

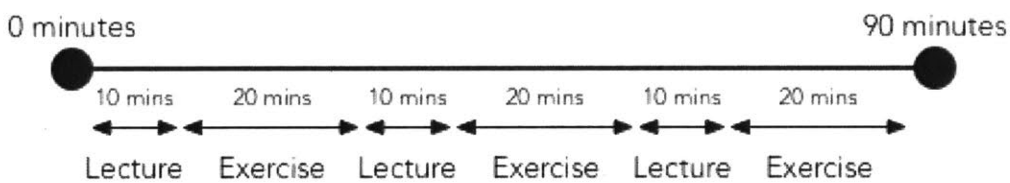


Figure 3-5: Lecture & Exercise Agenda

3.5.3 Homeworks

The homework assignments are distributed on a weekly basis and need to be handed back within a weeks time. Distribution, submission, and grading are done using the trio of GitHub, Travis-CI, and the Node application. Similar to the exercises, the homeworks start off with a boiler code template and a set of instructions to help the students either pass the necessary tests or view the required visual feedback. As of mid-April a total of one non-graded and five graded were assigned. Tentatively, a few more assignments were scheduled, but they were subject to the constraints on the effort required for the project which was due to start soon.

Table 3.2: Homework Schedule

Homework	Topic	Tag
Homework 0	Syntax	intro
Homework 1	Pac Mac	fundamentals
Homework 2	Binary Clock	fundamentals
Homework 3	City Data	data science
Homework 4	Course Catalog	data science
Homework 5	MNIST	machine learning

3.5.4 Quizzes

Two-term quizzes ensure that the students are able to formalize their learnings and develop a good understanding of the concepts of each of the topics discussed in class. Unlike traditional methods, the quizzes are open book, open computer, and open internet. Distribution, submission, and grading are done through the same method used for the homeworks and exercises. Quiz results and feedback will be discussed in a subsequent chapter.

Table 3.3: Quiz Schedule

Quiz	Tag
Quiz 1	fundamentals, data science, software architecture
Quiz 2	machine learning, web, iot

3.5.5 Project

The project is the final entity of the required course work. The last three to four weeks are exclusively reserved for work on the project during the in class meetings. This allows students to mark a dedicated time and place for the project group to meet, work and discuss any impending matters related to the project. Students are encouraged to pick problems related to the world of data science, machine learning along with some sort of a web or internet of things component. Past projects have included identifying patterns of terrorism, building an autonomous rescue operation bot, using a voice-controlled assistant to control light switches, etc.

Table 3.4: Project Schedule

Project	Tag
Project 0	data science, machine learning, web, iot

The grading process for the project was more of a qualitative effort requiring manual inputs from the teaching staff. All in all the project was broken up into four deliverables as described in Table 3.5. Each deliverable was to be packaged as a YouTube video and graded according to the rubric described in Table 3.6

Table 3.5: Project Deliverables

Deliverable	Description
Deliverable 1	Introduction of team and project
Deliverable 2	Planning of architecture and technological details
Deliverable 3	Working demo of the project
Deliverable 4	Final submission with core project work

Table 3.6: Project Grading Rubric

Measures	Points
Complexity and Sophistication	15
User Interface	15
Effort	15
Design	15
Video & Presentation	40
Bonus	10

3.6 System Design Architecture

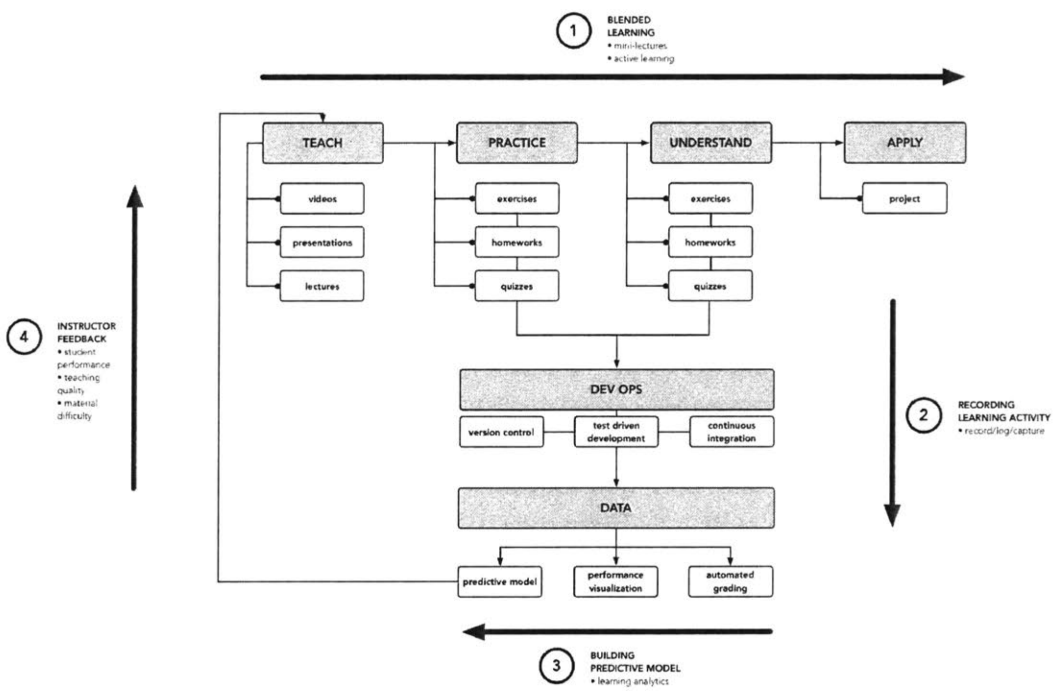


Figure 3-6: System Design Architecture

In Figure 3-6, we describe the entire system architecture design setup for the course. The blended learning philosophy for the course stems into four tiers; teach, practice, understand, and apply. The teaching tier comprises of video tutorials, presentations and classroom lectures. The practicing and understanding tiers are divided between the active learning exercises, homeworks, and quizzes. Finally, the applying tier consists of a month long group project where students apply their learnings to create an original piece of working software. The individual modules in each of the tiers comprise of a mix of face-to-face interactions plus online interactions. The tiered model thus represents a good baseline for a blended learning model.

The learning analytics portion in the system is through the use of applying the DevOps philosophy for the exercises, homeworks, and quizzes. Each of the assignment activities is completed using a combination of version control, test-driven development, and continuous integration. The submission results are captured and injected into a data pipeline that powers a predictive model, performance visualization dashboards and also to perform automated grading.

Lastly, the loop between the blended learning model and the learning analytics component is built through the feedback produced by the predictive model. The feedback consists of information regarding student performance which in turn could be used to modify teaching quality and difficulty levels.

3.7 Class Collaborators

The class was the result of a collaborative effort through the inputs of an over half a dozen individuals serving in various capacities. The hierarchy is shown in Figure 3-7

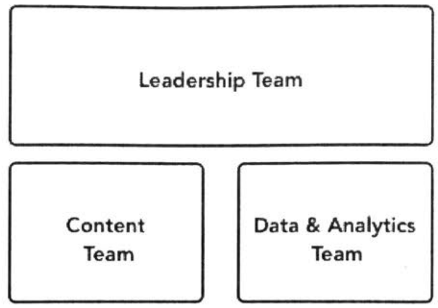


Figure 3-7: Class Collaborators

- Leadership Team - The leadership team consisted of the faculty members who were responsible laying down the agenda of the class and leading classroom discussions. Material for the classes would be prepared by the faculty and distributed to the larger team for feedback.

- Content Team - Once a high-level agenda for the curriculum was set up, the content team comprising of postdocs, graduate, and undergraduate students, was responsible for creating exercise and homework material to be used for active learning purposes.

- Data and Analytics Team - The data and analytics team's main role was to create the infrastructure to support the capturing and storing of assignment data as well as process the data to find useful metrics and insights.

3.8 Corporate Sponsors

The course was a recipient of a generous grant given by Google Cloud and GitHub. The grant was in the form of redeemable credits, and they were useful in helping setup the infrastructure of the course at a reasonable cost.

Chapter 4

Infrastructure and Data Pipeline

4.1 Purpose

A blended learning model must include a robust infrastructure to support the course activities. In addition, the implementation of a reasonable data collection process to measure learning analytics must also be designed with the particular context in mind. The two main components that have been identified for establishing a reasonable baseline for meeting the objective of developing a sound blended model and learning analytics module are shown in Figure 4-1. The first principle is around access to content which includes accessing content for learning purposes like presentation, slide decks, videos, and other artifacts as well as testing material that include exercises, homeworks, and quizzes. The second aspect is around delivering sufficient feedback back to the source in the form of scores, hints, and areas of improvement. A feedback of this kind can only be established when there is a good exchange of data between the students and the instructors. Such an exchange can be facilitated using a data capture method followed by a data analysis method.

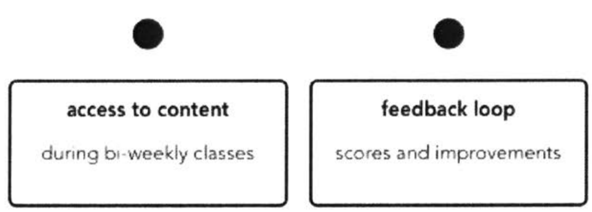


Figure 4-1: Baseline Components

4.1.1 Access to Content

The course has a dedicated website set up at http://www.onexi.org as seen in Figure A-1. The website acts as a one-stop shop for all content that has been created for the course. The various tabs on the navigation bar lead into the content zones for the lectures and homeworks. Each classroom meeting has a mini-lecture component and a series of active learning exercises. To help prepare the students for discussion, a series of byte sized video lectures help fill the potential gaps in their knowledge. A screenshot of an in use class session is shown in Figure A-2. Homeworks and Quizzes follow a similar model. The majority of the videos are hosted on YouTube, and the exercises, homeworks, and quizzes are distributed through GitHub. A snippet of an exercise on data manipulation hosted on GitHub is shown in Figure B-1.

4.1.2 Feedback Loop

An essential element in the blended learning cum learning analytical model was to create a feedback loop such that both students and instructors had an acute sense of their progress as the course progressed through the semester. Each submission of each exercise, homework or quiz had to have a feedback loop that guaranteed a score regarding the well-being of the students. We make use of the principles of version control, test-driven development and continuous integration to achieve the purpose of providing feedback back to both the students and instructors. For the students, the initial feedback is in the form of code hints using an IDE, and local testing was done using a test suite provided as part of the assignment. For the instructors, the testing data is collected and visualized in a web application that receives a stream of real data from ongoing assignment activities. The feedback loop system thus created is depicted is Figure 4-2

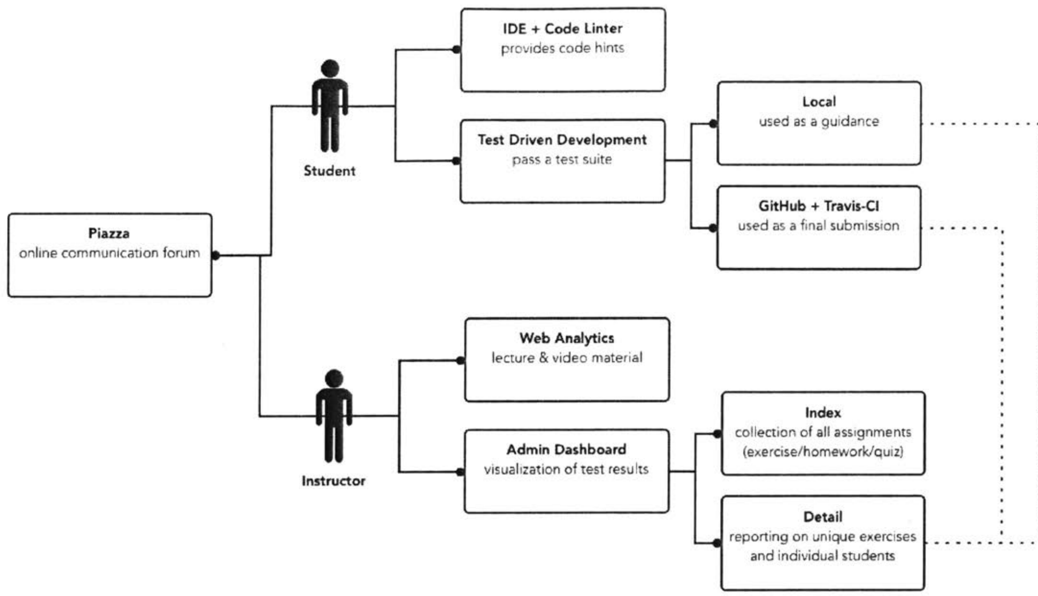


Figure 4-2: Feedback Loop between Students & Instructors

IDE and Code Linter

An IDE stands for an Integrated Development Environment and provides all the essential tools necessary to handle code editing functionality. IDE come in many shapes and forms, but for the course, we made use of Microsoft Visual Studio due to its cross-platform compatibility and its platform like nature that has allowed third-party developers to build plug-ins for the editor. In Figure C-1, we see the editor with an open exercise file along with feedback from the code linter suggesting that the variable is not being used in the function. The linter highlights missing keyword statements, incorrect syntax declaration and in some cases even certain logical issues.

Unit Testing

The idea of test driven development first came during the conversation around Agile software development. The kind of testing followed here could be best described as unit testing. The assignments are structured in a form such that each task can be broken down into a singular function. Each such singular function must meet the requirements and expectations of a corresponding test case that comes along with the assignment. An example is shown in the appendix where we have the functional exercises in Figure D-1, and the corresponding test case that must pass on execution in Figure D-2. The feedback that such a system displays on execution is shown in Figure D-3. In this case, we see that both the tests have passed, but in the scenario that a particular test has not achieved its intended outcome, the relevant feedback would be displayed back.

A secondary check is initiated when the assignment is submitted back to GitHub through a pull request mechanism. On receiving the pull request, an external webhook [1] instructs the Travis-CI system to run a final check on the submitted assignment and ensure that they pass the corresponding test cases. On a successful passage, the system reveals a green sign on the GitHub pull request page as seen in Figure D-4. These pull requests are considered to be a final submission of the student's work and are used for grading purposes.

Dashboard

The dashboard component is a real-time visualization tool for the instructors to view the output of the in-class active learning exercises, the weekly homeworks, and term quiz results. As shown in Figure **??**, the dashboard lists all the submitted exercises, homeworks, and quizzes that have been allocated. Each such assignment has an additional page that represents the execution results either from testing conducted locally or testing done as part of a final check by Travis-CI. The data collected

[1] A webhook is an event-driven API that informs a third party of an action that has taken the source platform

is broken down on two dimensions. The first dimension is based on the number of exercises that are part of the overall assignment. The second dimension breaks it down based on an individual student's performance allowing the instructors to get a sense of which exercises have been completed and which exercises have been a struggle to complete. Additionally, the individual performance scores help keep track of the general state of the class. The implementation can be seen in Figure ??

Piazza

An additional form of feedback loop was established through the use of Piazza, an online forum for communication on classroom activities. Piazza was exclusively used for the announcements made by the instructors. The students seeking clarifications on assignments would also post to Piazza. The open and collaborative nature of the platform allowed the entire class community to benefit from the conversations taking place on the platform. In Figure E-1, we see a student questioning the functionality of a particular module and the corresponding response by an instructor.

4.2 Publishing Assignments

4.2.1 Structure of Repository

Each assignment is structured as a unique independent GitHub repository. Each such repository comes with a starter code template that provides the skeleton of the code that must be filled up as part of the assignment solution. In addition, the starter code also includes the necessary configuration framework for running the test suites as well as the functional pieces to capture the performance data through both local and Travis-CI testing.

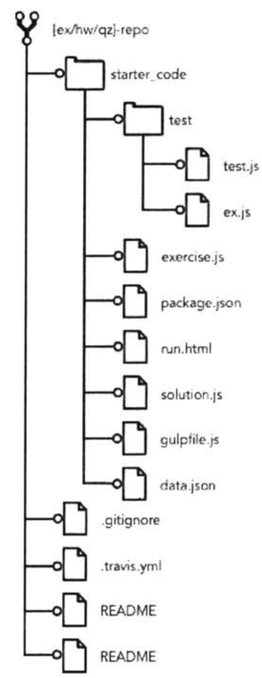


Figure 4-3: Repo Tree

- *starter_code*
  - *test*
    - *ex.js* - A solution to integrate the written exercises with the test suite
    - *test.js* - Collection of test suites that should pass on execution
  - *exercise.js* - Placeholder code for the assignment contained in here
  - *package.json* - The configuration file representing dependencies used in the assignment
  - *run.html* - An interface for the browser that integrated with the functions of the exercise for visual output
  - *solution.js* - The solution file to the exercise
  - *gulpfile.js* - Scripting file to enable capture of test run data
  - *data.json* - Potential store of data arrays or objects that are used as part of the exercise
- *.gitignore* - A list of files to be ignored by the git system
- *.travis.yml* - Configuration for running the submitted assignment on Travis
- *LICENSE* - Description of the license of the repository
- *README* - File for any particular instructions or miscellaneous notes

4.2.2 Steps to Publish

It takes a number of steps to get the assignment repo built up to a stage such that it is ready for publishing. We make use of an automated build process using Gulp to handle the intricacies for each of the steps above. The publishing flow is shown in Figure 4-4.

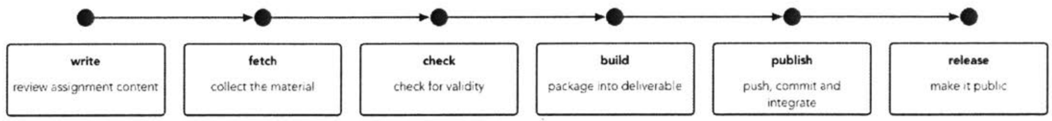


Figure 4-4: Steps

Write

The write step is a two-fold process where a particular assignment as developed by the content team is reviewed by the leadership team and packaged into a form that conforms to the template structure discussed earlier. By far it is the most manual step in the process. However, the benefit of doing so is to provide a context to the instructors on potential pitfalls the assignment could have when handed down to the students.

Fetch

At a date before a few days before the scheduled distribution of the assignment, the reviewed content is downloaded into a local folder and becomes the working directory for the forthcoming step.

Check

The local copy of the assignment is then checked to ensure that the dependencies have been well defined, the code is executable and all the cases defined in the test suite pass.

Build

The build step repackages the assignment but is devoid of any solution files. The output of the build step is a folder that conforms to the specifications of the initial template.

Publish

The publish step uses GitHub to create a new repository on the course account and commits the output of the build step. In addition, the publish step is also responsible for creating the necessary integration with Travis-CI.

Release

The previous step publishes the assignment as a private repository and finally, at the moment of the scheduled delivery, either a classroom meeting, a homework release date or just before a quiz begins, the built repository is made public to be accessible by the students working on the assignment.

4.3 Submitting Work

The submission process comprises of some commands that are within the purview of a traditional software development cycle using a version control system. The entire chain of commands is shown in Figure 4-5. The chain of commands can be broken into three distinct steps as discussed below.

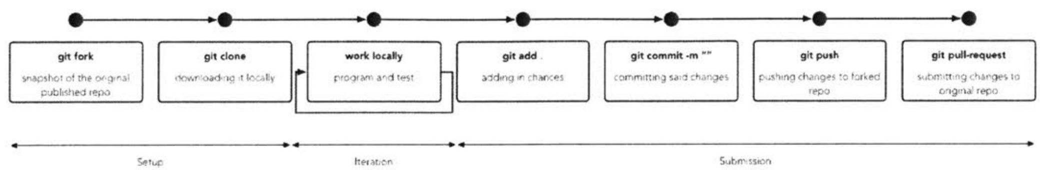


Figure 4-5: Submission Steps

4.3.1 Setup

The foremost step is the process involving the forking and cloning of the published assignment repository. This step essentially facilitates the creation of the snapshot of the original repository and helps to create a local copy of the snapshot on the computing machine being used for the assignment.

4.3.2 Iteration

Once a local copy has been created, students can keep on iterating on writing code and testing the code such that they can successfully pass all the tests cases defined in the test suite.

4.3.3 Submission

Having completed writing code for the assignment the students are then required to submit their changes back to the original repository. The first step in the process of submission is to add the modified files to mark them ready for committing. The next phase is to go ahead and commit the files and pass a message highlighting the nature of the changes. Up till this moment, the changes have only marked for commit on the local copy of the repository and the push command ensures that the local changes are uploaded to the original repository. Once the changes have been pushed, they are forever available on the timeline of the repository. The last step in the process is to merge the changes that are currently on the snapshot, to the original repository that was the foundation of the entire exercise. The last step is performed by creating a pull request of the submission code.

4.4 Ingesting Data

As shown in Figure 4-2, there are two types of data points that are captured by the automated testing framework. The first set of data points collected come in the form of test runs executed by the students locally while working on the assignments. The second set is collected through the tests run by Travis-CI once the final submission has been pushed to GitHub. However, both these collection processes work in the same way as highlighted in Figure 4-6.

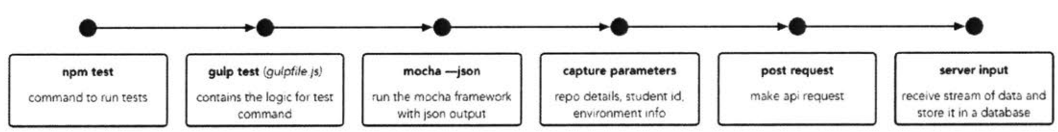


Figure 4-6: Ingestion Flow

4.4.1 Capturing Data

The data ingestion process is put into execution by running npm test, a simple two-word statement on the command line interface. The command starts a new process that invokes code written in the gulpfile which serves as the set of instructions for the data capture process. The code performs two steps synchronously. In the first step, it initializes the mocha framework and runs the test suite. The output of the test run is kept in memory while the code captures additional parameters that figure out what assignment is being worked on currently, identify the student running the test and lastly finding out the environment that the test is being run on (whether it's a local execution or a Travis-CI execution). Once all these data parameters have been captured, the system then constructs a single payload and makes an API request to a remote server. The remote server receives the stream of data which is stored in a database for further analysis.

4.4.2 Data Structure

A detailed sample of a data point is shown in Figure G-1. The four main key attributes are defined as follows.

- `task` - contains information regarding the name of the repository and the environment in which the test was run
- `auth` - contains information to tie back the test results to a unique individual
- `data` - include a breakdown of the test reports containing summary statistics as well as reasons for pass/failure of an individual test
- `lint` - primary objective was a capture the source code on which the test was run but also includes information on potential syntax issues

Chapter 5

Analytics and Predictive Model

5.1 Student Demographics

The students comprising the class hailed from a multitude of departments and education levels. The course membership as of early April 2017 stood at 48 students with 12 registered as students and 36 students taking it for credit. We anticipate the number of students taking it for credit would drop by around 10 to 15% owing to the scheduled course drop date in mid-April 2017. Based on a survey with 30 respondents, we were able to get an initial foray into the composition of the class and the level of experience the students possessed.

A majority of the class membership comprised of students at the graduate level Figure H-1. However, there was a good diversity in terms of the departments that the students represented Figure H-2. Management and Engineering were in the majority with a few Architecture majors forming the next big component. Most students were predominantly taking the course out of interest Figure H-3, however for a few undergraduate students the course was a mandatory requirement for their degree qualifications. A lot of the students came in with a certain understanding of programming concepts Figure H-4. However, almost most of them came in with no JavaScript experience whatsoever Figure H-5. Additionally, based on a few weeks of experience most students considered the course work to be at a medium level Figure H-6 and the amount of effort exerted by each student was a widely split number

Figure H-7. All in all, it was a diverse group of students hailing from a multitude of backgrounds and with an evenly spread out distribution of experiences.

5.2 Preliminary Analysis

Our first foray into analyzing the captured data shows a good balance of runs conducted on the local platform before it is submitted to the Travis-CI platform Figure I-1. The information indicates that students are thorough and diligent in their pursuit of running tests while working with the code. Once they've gathered good signals regarding their answers, the final push submission is made. A breakup of the tests by exercises, homeworks and quizzes show that the ratio of local runs to Travis-CI runs are the highest for the homeworks. The revelation suggests that the students take more time to digest the intricacies of reaching to the correct solution while working on the homework assignments Figure I-2. Note that the quizzes were structured in such a way that the local test suite was not available for the students. The quiz submissions would be graded on a final check done through the Travis-CI platform.

In Figure I-3 we chart out the percentages of students who score a full grade by passing all the tests contained in the assignment exercise. On the y-axis, we map out each assignment, and each unit on the x-axis represents a single run and the portion of students who get all passing tests. The x-axis is normalized with a max threshold of 10 runs such that we can capture the wide array of runs required for each exercise. We notice that earlier in the semester, students can get tests to pass with a fewer number of runs which could be attributed to the hands-on support that students were given in the initial days. As we progress towards the later half of the semester, the number of runs crosses the threshold of 10 minimum runs. We also see a relatively high percentage of students who can pass at least 80% of the tests.

Throughout the semester, we can witness a few cases where we get a 100% pass rate.

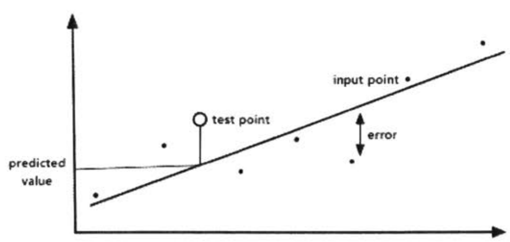
5.3 Prediction Strategy

The application of learning analytics in a blended learning model allows the possibility of predicting student performance. The peak into student performance would give the instructors a feedback loop to adjust the course content such that it is more favorable for positive student outcomes. The question then arises as to how do we go about creating such a model. Based on the work discussed in the previous chapters, the infrastructure supporting the course must have a dedicated data collection process. In the example of the course discussed earlier, the data capture was in the form of user attributes, and code execution runs. Based on the data thus collected, we could build a statistical or machine learning tool to help accomplish the purposes of prediction. What type of a prediction model could we create, what are the input features that could be fed into the model and what are the output values that must be obtained to provide valuable insight?

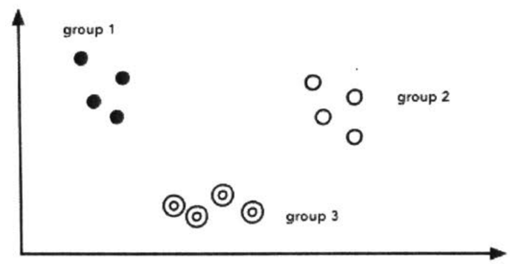
The model selection task is one that is faced with a plethora of choices. In the machine learning sphere, two parent classes of learning models are dominant; supervised and unsupervised learning Figure 5-1. In the supervised learning scheme, the input data is associated with an output label. Thus when the model is being trained the internal parameters are adjusted up till the point at which the input data leads to the output data with the least possible error for all the input data points. After having trained the model, a new input point that comes in with an unknown output label, when fed into the trained model will output a label value that would be considered as the prediction for the new input point.

Under the unsupervised regime, the input data has no notion of an output label.

Instead, the algorithms try to finding meaning based on just the input parameters. An explanation of a commonly used algorithm would be to think of comparing each input point with every other remaining input point. While comparing a pair of such points, the system works towards finding a similarity score between them. The points which turn out be most similar can then be grouped into one distinct category.



(a) Supervised Training



(b) Unsupervised Training

Figure 5-1: Learning Model

The choice of model eventually boils down to the specific use case that is being solved. In our case, we can observe the benefits of both these learning methods. In the unsupervised case, we would be able to group students into clusters with each cluster representing an average score, which in turn would be the determinant to identify students at the risk of failure. In the supervised model, we would be able to obtain a predicted final grade for each of the students based on a model trained with a sufficient number of training points. The grade thus predicted for each student would be an indicator of understanding their potential performance.

5.4 Algorithm Description

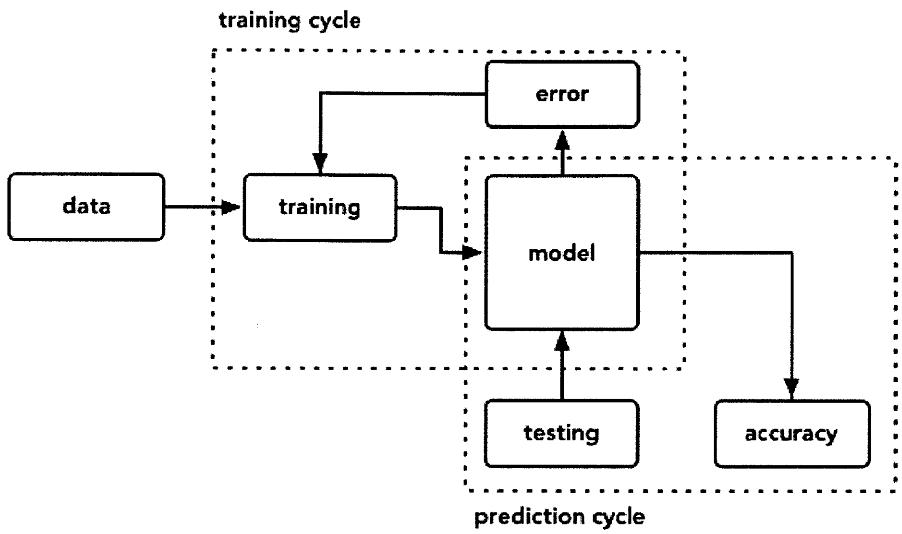


Figure 5-2: Supervised Learning Strategy

We make an attempt to build a predictive model using the supervised learning strategy Figure 5-2. In particular, we make use of an artificial neural network for creating the model. Neural networks have had a great impact in the field of machine learning in the recent past due to their ability to produce excellent prediction results. They've been widely known to have accomplished never seen before results in the areas of computer vision and natural language processing. They've also shown promising results in the areas of classification and regression. A large reason for their advent has been due to the proliferation of large quantities of data that can be processed efficiently using graphical processing units. Google Brain, Facebook AI Research, Uber ATC and Microsoft Research are just some of the entities within large tech corporations that have been putting in a lot of resources around the fundamental research of machine learning principles with a deep focus on "deep learning" techniques aka neural networks.

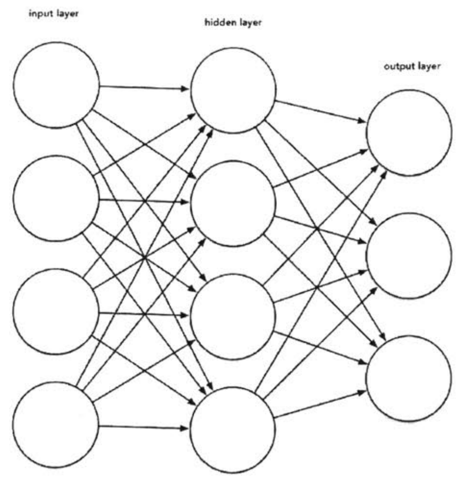


Figure 5-3: Neural Network Architecture

The basic idea of a neural network stems from the concept of having a network of perceptrons connected in a series of layers. A small network is shown in Figure 5-3. Each perceptron in a particular layer receives a set of inputs from the previous layer. These inputs are then aggregated using a bunch of weights and activated using an activation function. Some of the most common activation functions are sigmoid, tanh and ReLU. The input layer is represented as a layer of neurons that simply propagate the input values to the next layer. The output layer, in a classification problem, would have k neurons, each one representing the outcome of being in class k. What we have described so far is the feed forward step, where the inputs are propagated through the multiple layers to get an output in the last layer.

The feed forward step is followed by the backward propagation step, where the essential idea is to modify the weights in each layer by taking the gradient of the propagated loss with respect to the weights in that layer. Thus the contrast between the feed forward and back propagation step lies in that the former involves translating the weights to the output layer, whereas the latter consists of translating the loss in the output layer to the weights in the previous layers.

A theoretical formulation using an example of a 2 class classification problem solved using a neural network with a single hidden layer comprising of 3 perceptrons is given in Appendix J. Each perceptron in the hidden layer consists of a ReLU activation function. The output layer consists of 2 perceptrons and has an activation function defined by the softmax function. The loss error in the last layer is calculated as the partial derivative of the loss with respect to the aggregation in that layer. Since the loss error is a cross entropy loss function, the derivative of the cross entropy with respect to the aggregation is essentially the activation minus the targets, where the targets are part of an one hot vector. We can easily generalize the above derivations by replacing the last layer (2) with (L) and the subsequent inner layers with (l). We can also assume k output layers and d input layers. Applying the recursive rule after that results in transforming the mentioned derivations into an algorithm that works for any deep neural network. Also, note that η is the learning rate which can either be constant or scaled as a function of the number of epochs. The bias terms, ignored above for simplicity, can be simply added into the aggregation terms in each layer. During the update rule, the bias is $\delta^(l)$ for that particular layer. We denote the configuration of the neural network using $(m_1, m_2 ... m_N)$, where N is the number of hidden layers and m_N refers to the number of neurons in the n^{th} hidden layer.

5.5 Identifying Features

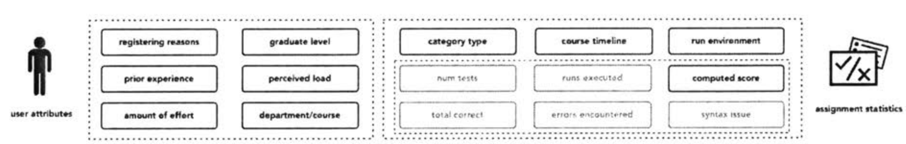


Figure 5-4: Potential Features

In Figure 5-4, we show a list of potential features that have been identified as an input to the model. The features are broken up into the two sources from which they are obtained. The user attributes are obtained from a survey sent out to the students. The user attributes are taken as is from the options provided in the survey. The assignment statistics are pulled out of the individual runs. Note that assignment statistics are measured for every permutation of a student and assignment.

One of the assignment statistic features is the score attribute. The score takes into account the total number of runs that were executed, the total number of tests that passed and any potential errors or syntax issues encountered. The purpose of calculating the score is to reduce the total number of points for each submission into a singular point that still captures the underlying metrics associated with the submission. The formula used for calculating said score is defined below. We loop through each of the individual runs for an assignment. For each such run, we calculate the delta of the number of tests that pass in that run and weight it by a decreasing function dependent on the run number.

$$score = \sum_{i=1}^{n} r_i$$

$$r_i = \left(1 - \left(\frac{i}{a}\right)^2\right) * \left(\frac{cur\_correct - prev\_correct}{total\_tests}\right)$$

$$\text{where } i = run\_number$$

Eventually, the input matrix would resemble the one shown below. Each data point represents one assignment submission made by one user where the individual runs have been summarized in the score parameter. The details of the different attributes are shown in Table 5.1

$$\begin{array}{c} \\ input_1 \\ input_2 \\ \vdots \\ input_n \end{array} \begin{array}{c} \begin{array}{cccccc} attribute_1 & attribute_2 & attribute_3 & \ldots & attribute_m \end{array} \\ \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1m} \\ r_{21} & r_{22} & r_{23} & \cdots & r_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{n1} & r_{n2} & r_{n3} & \cdots & r_{nm} \end{pmatrix} \end{array}$$

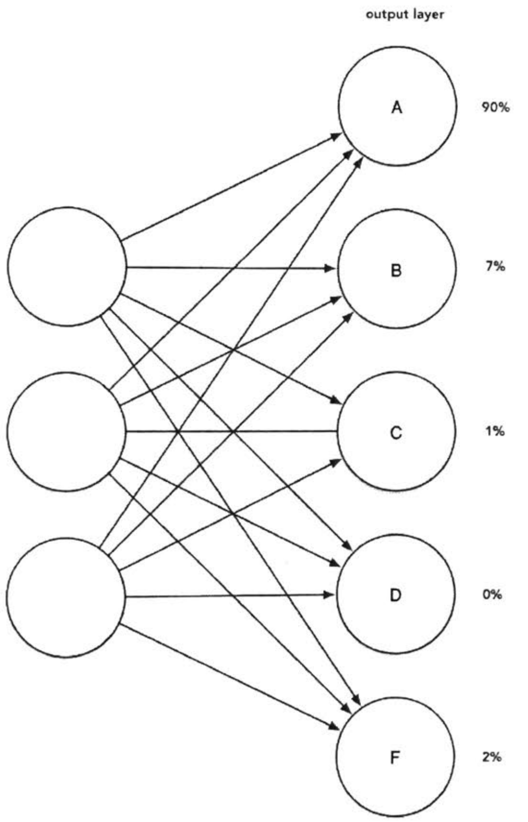
Table 5.1: Attribute Descriptions

Attribute	Name	Range
$attribute_1$	User Programming Experience	0 - 5 years
$attribute_2$	User Department	1 - engineering, 2 - management, 3 - architecture
$attribute_3$	User Level	1 - undergrad, 2 - grad
$attribute_4$	User Effort	0 - 20 hours
$attribute_5$	User Load	0 - low, 1 - medium, 2 - high
$attribute_6$	Assignment Type	1 - ex, 2 - hw, 3 - qz
$attribute_7$	Assignment Timeframe	1 - 12 weeks
$attribute_8$	Assignment Score	0 - 100 points
$attribute_8$	Assignment Environment	1 - local, 2 - travis

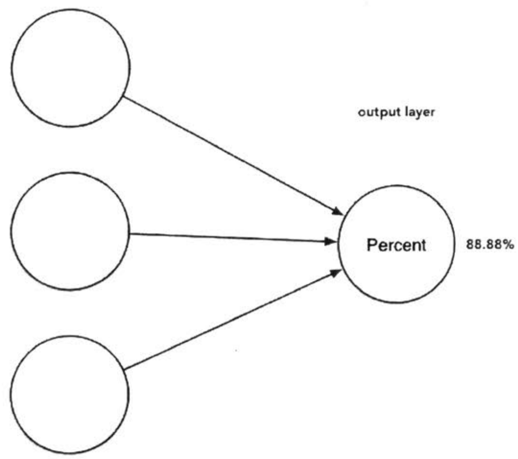
5.6 Output Classes

There are two possible choices of outputs that could be obtained using the model. Using a regression model, the output could be a score between 0 - 100. A higher score would be assigned to an individual who is perceived to be performing well in the class Figure 5-5. The scoring method would be considered a regression problem and has the advantage of obtaining a more granular score for the students. Another possibility is using the grades A, B, C, D & F as the output labels. The problem is then reduced to a classification problem and is well suited for a neural network.

As shown in Figure 5-5, the classification problem thus requires a neuron for each of the grades in the output layer. The network then produces a value for each of the output neurons which signifies the probability of the input point belonging to that class of the output neuron with a certain degree of confidence.



(a) Output Grade Probabilities



(b) Output Percentages

Figure 5-5: Output Labels

5.7 Training And Testing

In Figure 5-6, we highlight the system architecture and design flow of the neural network model we aim to create. As described earlier, the supervised learning strategy requires a set of input points that are used as a training mechanism for the model. Each such input point for our model will be an object containing a set of user attributes and assignment statistics along with the final grade of the user. Such input points are generated for each unique permutation of a student and an assignment. The final grades are obtained from the grade book that is kept in check by the teaching staff. Once the model has been trained with a sufficient number of input points, any new incoming input points which have not been labeled with a final grade, can obtain a predicted final grade from the model.

The training of a neural network model is a difficult process that can be considered to be more of an art than a simple logical decision-making process. The design of the architecture of the neural network model is a paramount first step that has great significance in the overall accuracy of the model. There are some parameters that can be regulated to create a good model. As part of the training process, we use a number of permutations of configurations to build an appropriate model. For automation of the various cases, we use a grid search cross validation process. Such a process automatically creates permutations based on the possible parameter values set for each of the attributes. For each such permutation, the system then splits the training set into a number of sub training and validation sets and executes the training process for each such set. The average accuracy or error rate, based on the validation set, is then determined for the configuration and this process continues for all the permutations that were initially created. In the end, the permutation with the highest accuracy or least error is picked out to be the suitable model.

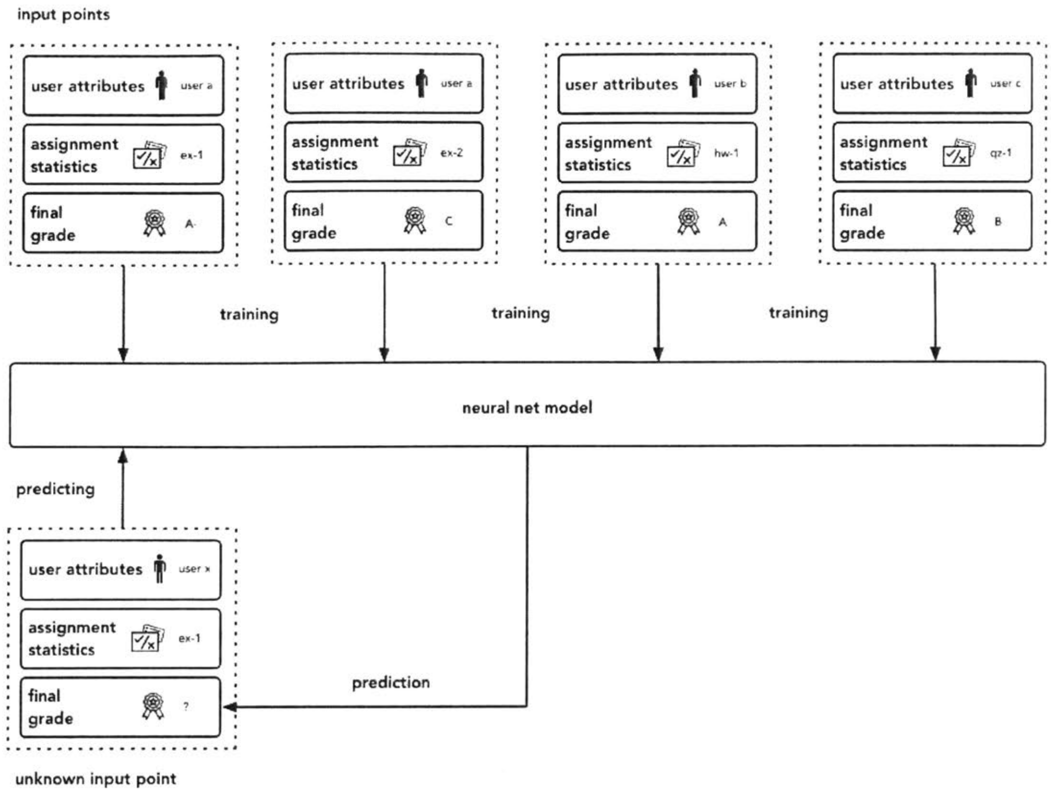
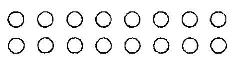


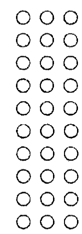
Figure 5-6: Neural Network Based Model

There are a number of different parameters that can be adjusted while training a neural network model. In addition, a number of techniques can be adopted to mitigate any potential devaluation of the model while the training process is ongoing. Both these considerations improve the effectiveness and efficiency of the training procedure and have been widely used in the broader research community.

- Network Architecture - Network architecture refers to the construction of the individual layers of the neural networks. There are wide neural networks and deep neural networks. Wide neural networks have a larger number of layers while deep neural networks have a larger number of neurons within each layer. Recent developments have seen the proliferation of deep neural network, hence the trend word "deep learning". Deep neural networks have been considered to be extremely valuable in solving prediction problems in the fields of imaging, vision, language and speech.



(a) Wide Architecture



(b) Deep Architecture

Figure 5-7: Network Architectures

- Weight - The initial weights must be picked to ensure hat they do not they kill the gradient. When the gradient is 0, during the back propagation step, the network is unable to learn new weights. For a sigmoid, this can easily happen when the sigmoid function takes on values near 0 or 1, a condition termed as saturation. Hence, in order to avoid saturation, the weights must be picked such that they do not take on very large values. Therefore it is wise to randomize the weights not just with a zero mean and random variance, but with a variance that is proportional to the number of neurons in the previous layer. This ensures that the weights do not take large absolute values, thereby keeping the sigmoid outputs within a range that does not lead to saturation.

- Learning Rate with Decay & Momentum - An optimal learning rate is crucial for the network to converge to stable weights. If the rate is too low, the weight update rules wont see much of an improvement on successive iterations. If the rate is too high, the gradient will overshoot and blow up the weights. A solution to this problem is to have a variable learning rate that decreases over time as a function of the number of iterations. In conjunction with the learning rate decay, another theory that has been popularized is the idea of momentum[25]. Stochastic gradient descent is formulated by updating the weights using the gradient of the loss function scaled by a learning rate. Such a process could potentially lead to a slow learning process when the gradient continuously changes direction to account for inflection points. In such cases, when we apply the idea of momentum, the gradient continues to move in the direction of an accumulated average of its previous gradients.

$$\eta = \frac{\eta_0}{1 + \alpha t}$$

$$v = \mu v - \eta \nabla_\theta L(f(x^i; \theta), y^(i))$$

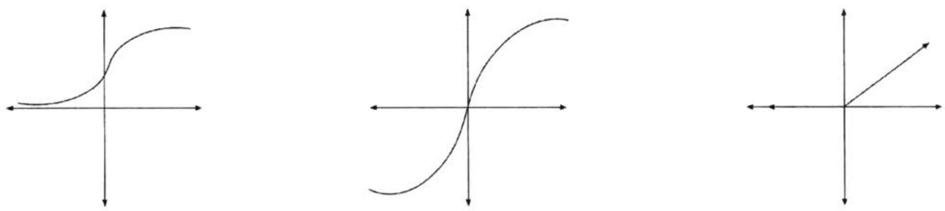
$$\theta_{t+1} = \theta_t + v$$

- Objective Function & Activation Function - The objective function is the error function that is used to find the difference between the predicted outcome and the correct value. In a case of regression, the objective functions could the Mean Squared Error or the Mean Absolute Error. In the case of classification, we make use of the Cross-Entropy Error. The activation function applies to each of the neurons contained in the network. Popular choices of functions are the Tanh, ReLU or Sigmoid functions.

$$L = -\sum_{i=1}^{2} t_i \log(a_i^{(2)})$$

$$L(f(x^{(i)}; \theta), y^{(i)} = \frac{1}{2}(f(x^{(i)}; \theta) - y^{(i)})^2$$

$$L(f(x^{(i)}; \theta), y^{(i)} = |f(x^{(i)}; \theta) - y^{(i)}|$$



(a) Sigmoid (b) Tanh (c) ReLU

Figure 5-8: Activation Functions

- Regularization - Overfitting is a problem in machine learning where the model is biased towards predicting good values for the training data but perform badly on never seen before inputs. Regularization methods are useful to avoid overfitting the training data. In order to prevent the weights from taking on large values, a regularizer term can be added to the objective function. The term would ensure that the weights are suppressed to the extent that the entire model does not exclusively do well only on the training data but is also effective on never seen before data. The bias-variance tradeoff [1], where we try to reduce

[1]Low Bias means less assumptions about the input data; less variance means lower divergence between the predicted output and correct output on never seen before data

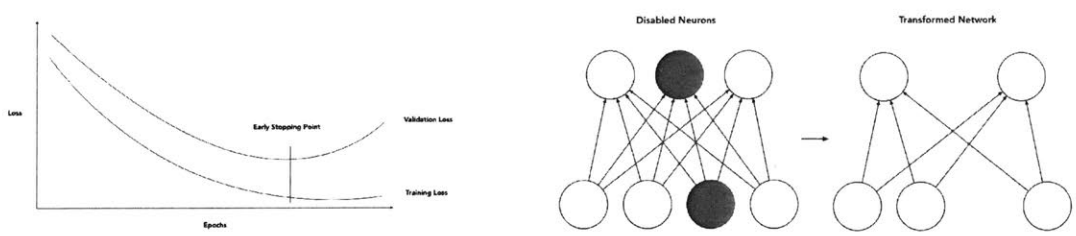
the bias and variance error can be adjusted with the use of the regularizer term.

$$L = -\sum_{i=1}^{2} t_i \log(a_i^{(2)})$$

$$J = L + \lambda(||W^{(1)}||_F^2 + ||W^{(2)}||_F^2)$$

L is the loss function and J is the loss function with the regularizing penalty

Early stopping is another regularization technique which involves stopping the learning process at a point where the validation score has not improved by much over the last few epochs. Dropout[24] is another technique used to regularize the network parameters. At each iteration of the feed forward step, certain neurons are disabled with a probability p such that their activated values are 0.



(a) Early Stopping (b) Dropout

Figure 5-9: Regularization Techniques

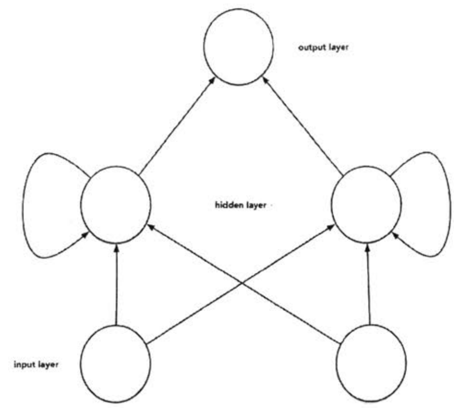
5.8 Implementation Improvement

In the model described previously, each individual training point is devoid of any information regarding previous performance measures. The only relevant connection to overall performance is through the final grade that is used as an output label for each of the input points. Within the neural network research community, the problem of the inability to incorporate past temporal data into the model was an important gap that needed addressing. The solution to the problem was developed in the form of Recurrent Neural Networks or RNN's. A RNN is similar to a feedforward neural network except for the fact that the hidden layers now not only accept the inputs from the previous layers, but an additional input is the output of the same neuron Figure 5-10. The input equation for a neuron now contains an additional variable that represents the output of that neuron in a previous time step. The structure thus allows the network to hold on to a certain form of memory from the previous outputs and is capable of building predictive models for data points that conform to a time defined sequence. These input sequences could be handwritten texts, speech clips, real-time video streams. The output results could be a prediction of the next perceived state of the input sequence.

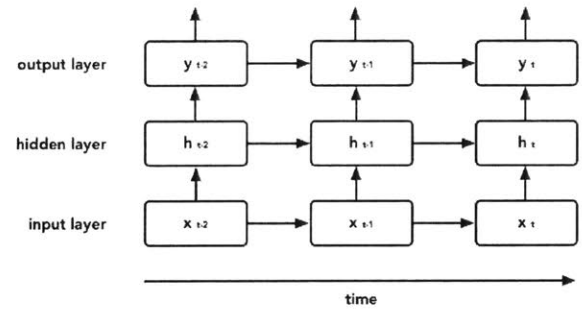
$$h_t = W^X.X_t + w^h.h_{t-1}$$

$$y_t = \sigma(h_t)$$

The one limitation with RNN's is related to the problem of vanishing gradients. The output values are computed using a series of feedforward and backward propagation steps. In the backward propagation step, the error feedback is sent back to the previous layers using a computation of the gradient of the error and this error is now compounded due to the presence of an additional input that is present in the form of the output of the neuron. The resulting model thus is not able to remember states from a time period from far long ago [2]. A solution to the problem of vanishing gradients was developed in the form of Long Short-Term Memory networks [13]. In



(a) Architecture



(b) Sequence

Figure 5-10: Recurrent Neural Network

a LSTM network, the architecture is modeled through a composition of cells. Each cells is equipped with a neuron along with a bunch of logic gates that can control the on/off states of the inputs and outputs Figure 5-12. These gates are operated by passage through a sigmoid neuron which outputs a value between 1 and 0 to signify how much of the previous data should be passed on. There are three gates in a LSTM cell, input gate, output gate and the forget gate. The three gates work in tandem to ascertain how much of the previous state and new input should be passed and stored as the current state. It is through this mechanism of being able to store past history that LSTM's exhibit a powerful capability of remembering data.

$$g^t = \tanh(W_g^X.X_t + w_g^h.h_{t-1}) \tag{5.1}$$

$$i^t = \sigma(W_i^X.X_t + w_i^h.h_{t-1}) \tag{5.2}$$

$$f^t = \sigma(W_f^X.X_t + w_f^h.h_{t-1}) \tag{5.3}$$

$$o^t = \sigma(W_o^X.X_t + w_o^h.h_{t-1}) \tag{5.4}$$

$$c^t = f_t \odot c_{t-1} + i_t \odot g_t \tag{5.5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{5.6}$$

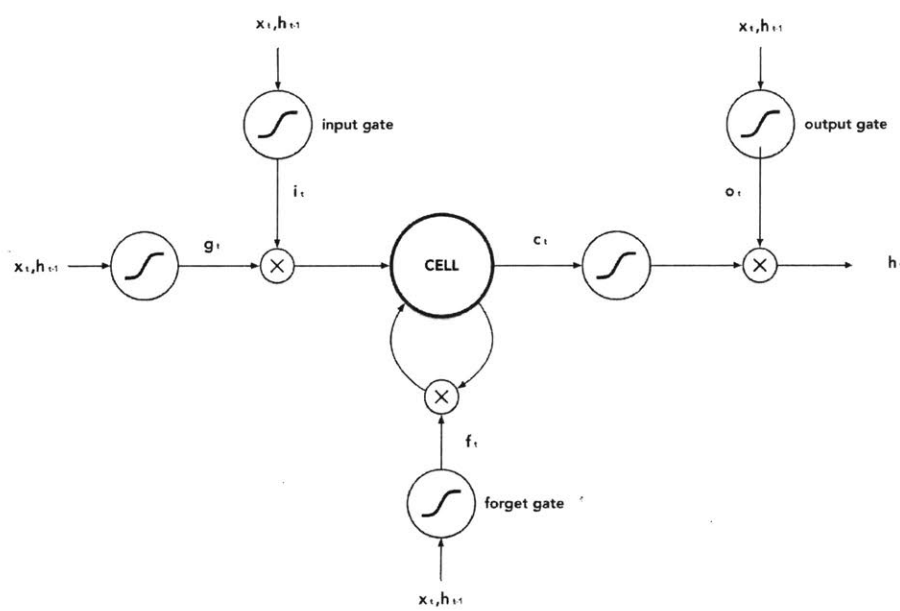


Figure 5-11: LSTM Network Cell

In our study of assignment results, we can use a LSTM network to predictive future progressive assignment results based on historical data. As highlighted in Figure 5-12, our sequence of inputs is composed of individual assignment statistics for each unique user. The output label for the said assignment statistic is set as the grade for the next assignment in line. Hence our network architecture uses data from the historical steps to be able to predict the grade outcome for the next assignment. Thus, when a new student comes in and submits an assignment, we would be able to predict the grade outcome for subsequent assignment. The predicted outcome would act as a good indicator to pull levers in the feedback loop to intervene in case the outcome is not up to the expected standard.

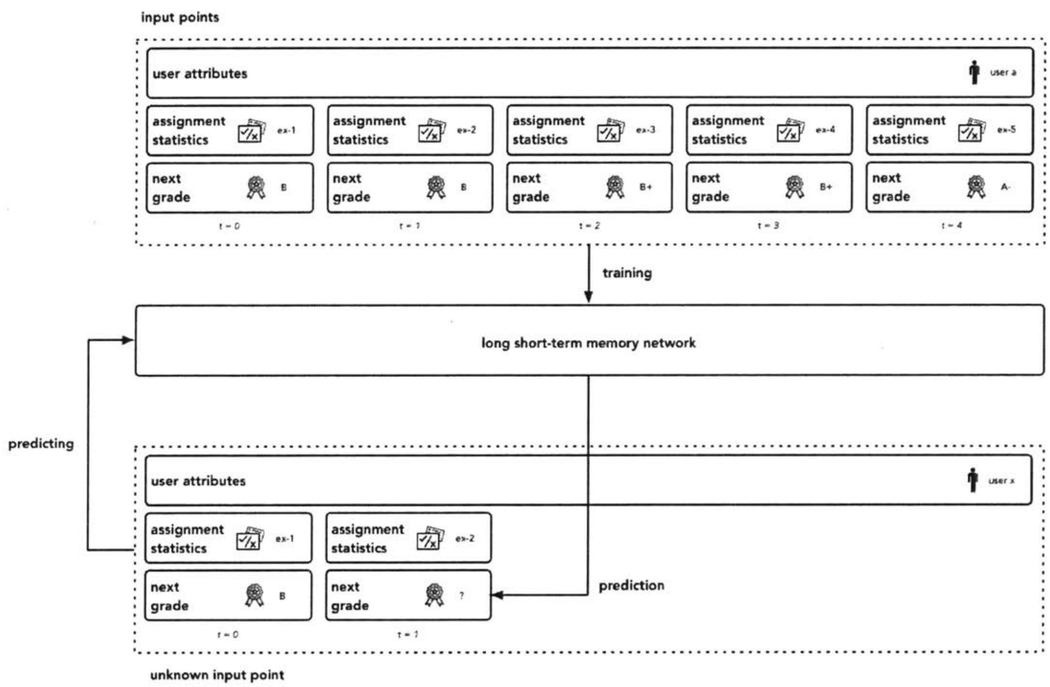


Figure 5-12: LSTM Based Model

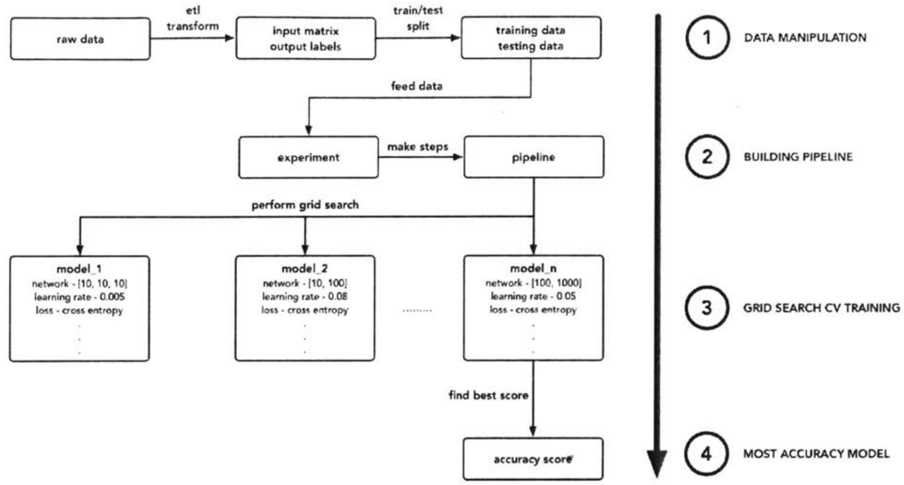


Figure 5-13: Training Mechanism

5.9 Results and Accuracy

We implement the neural network model using Python and Keras [2]. The code flow is highlighted in Figure 5-13. In fact, the predictive process is broken down into a series of four steps. In the first step, we clean up the data and prepare the data such that it is suitable as an input matrix for the model we will create. In the second step, we build the pipelining interface that will be able to serve as the foundation of our experiment. We then add in a number of models to the pipeline, each of which has been modified for a number of parameters. This lets us perform a grid search cross-validation on the model to eventually give us to most effective model with the best prediction score. In the next few sections we run a number of experiments to witness the effects of modifying various parameters while training the neural network model. Note that we run the experiments using the regression model of the output

[2]Keras is a deep learning library that provides easy to function by building an API on top of TensorFlow

labels and hence the score performance is measured using the mean squared error (MSE). A lower value of the MSE represents a better performance.

Network Architecture

The first experiment involves comparing the effects of a wide vs deep neural net architecture. A higher number of neurons does lead to better performance as can be seen in Table 5.3. However, extended the network to be wider does not lead to any added performance benefit. The diminishing returns can be attributed to the architecture failing to adjust the weights to an optimum level due to the presence of a large number of neurons.

Wide Arch	**MSE**
1*[100]	0.004
2*[100]	0.012
3*[100]	0.013
4*[100]	0.013

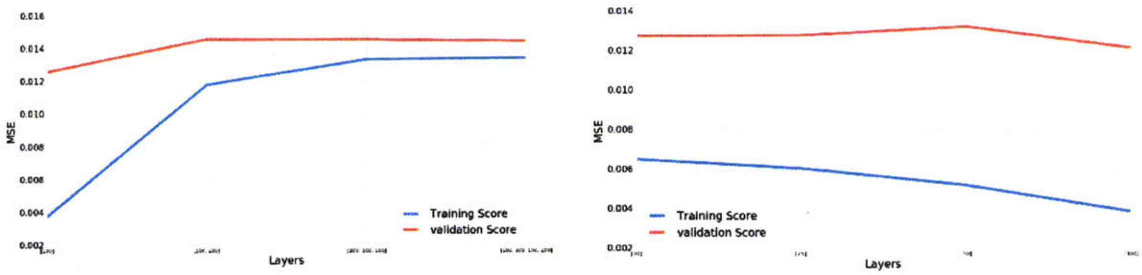
Table 5.2: Wide Arch

Deep Arch	**MSE**
1*[10]	0.006
1*[25]	0.006
1*[50]	0.0055
1*[100]	0.004

Table 5.3: Deep Arch

Number of Epochs

As seen in Table 5.4, a greater number of epochs results in better a lower MSE score. Since the network is trained for a longer number of iterations, it is able to adjust its weight to learn the data inputs in a more effective manner. This is large attributed to the backpropagation algorithm which makes of the stochastic gradient descent



(a) Wide Arch vs MSE (b) Deep Arch vs MSE

Figure 5-14: Architecture vs MSE

(SGD) optimizer. The SGD optimizer back propagates a small change in the error to the previous weights. These incremental changes are more widespread when the network gets a chance to train for a larger number of iterations.

Epochs	MSE
10	0.014
50	0.014
100	0.011
200	0.008
500	0.006
1000	0.004

Table 5.4: Epochs

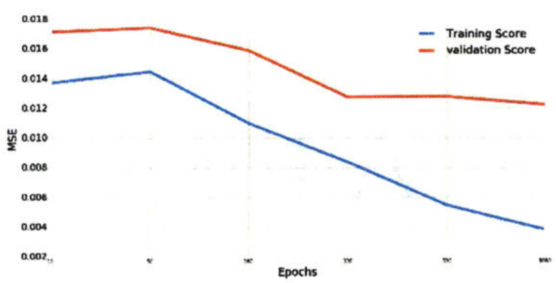


Figure 5-15: Epochs vs MSE

Learning Rate

The learning rate is used to control the speed at which the gradient errors are propagates back to the weights. A higher learning rate does lead to better performance but the performance peaks at a certain threshold beyond which the network sees diminishing returns on increasing the learning rate. The reason why this would happen is due to the learning rate being too large such that the weights overshoot themselves during the gradient descent. Thus the weights are not able to converge to the local minima and lead to a worse performance.

Learning Rate	MSE
0.001	0.011
0.005	0.009
0.010	0.008
0.025	0.006
0.030	0.004

Table 5.5: Learning Rates

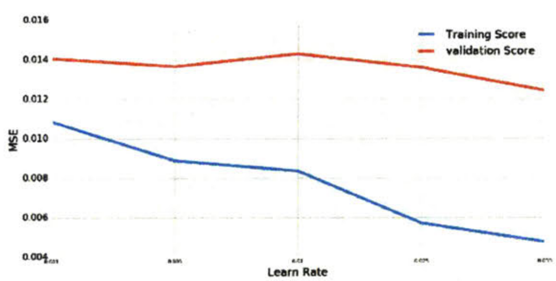


Figure 5-16: Learning Rates vs MSE

Chapter 6

Conclusion and Future Work

In this study, we have outlined the description of a blended learning model coupled with a learning analytics module to create a course on introductory programming and web applications. Our model comprises of a cycle between engaging in a blended learning model, capturing data from student assignment submissions, applying a predictive model using the data collected and finally funneling the feedback back into the blended learning model to adjust the course strategy for improving student outcomes Figure 6-1.

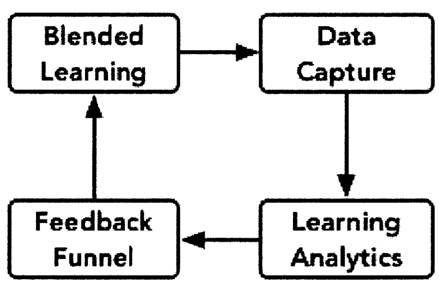


Figure 6-1: Course Model Cycle

Based on the analysis performed, we were able to identify some shortcomings with respect to the process for structuring the course content and data collection process. The following recommendations would aid in further improving the model for the course.

- Logging Code - The course makes use of the Visual Studio IDE which allows developers to write custom plugins for the editor. In order to capture the state of the code at each and every character change or compilation command, a custom plugin could be created by the course staff and have it installed on every student's computing platform.

- Comment & Feedback - An additional input source could be in the form of comments that are answerable by the students. The comments data could then be extracted for particular sentiments which in turn could then be used as a feature in the model constructed.

- Web Analytics - Another potential source of input data would be the data captured through the web. The main course website hosts lecture slides + mini videos. Using the data stemming from the interactions of the students with the course material could be another valuable source of input data that could be feed into the model.

- Designing Content - The course content must be further designed to cater for a blended learning model. Issues such as difficulty levels, breadth of topics covered, adherence to a time-boxed duration, ensuring assignments relevancy and testing for a predefined rubric of skill must be strictly followed.

Appendix A

Course Website

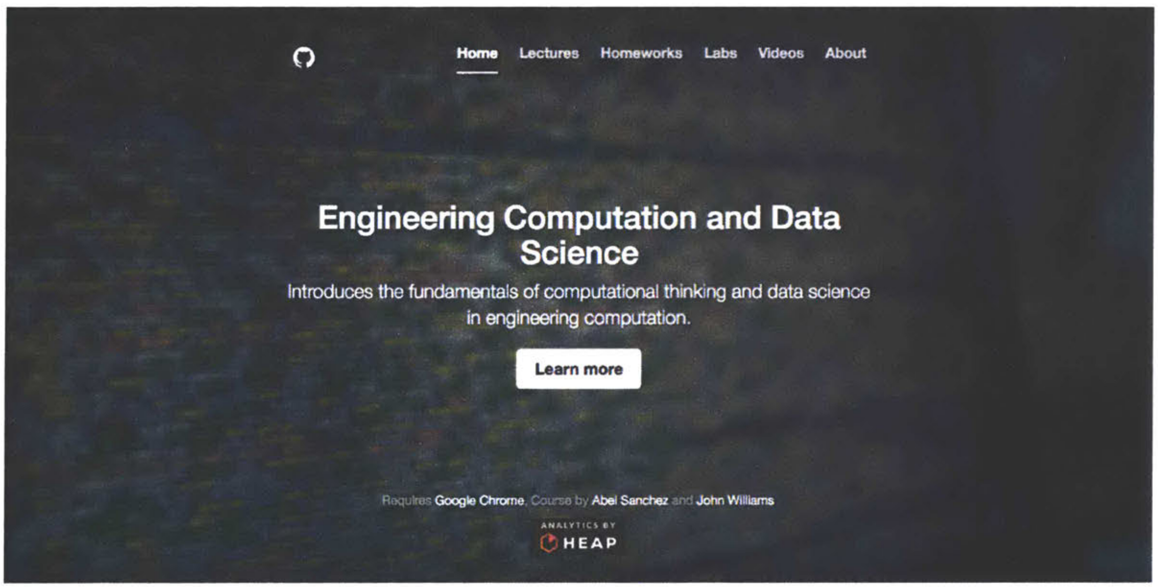
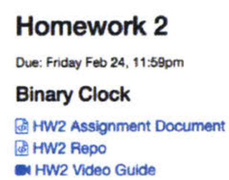


Figure A-1: Course Website

(a) Class Material



(b) Homework

Figure A-2: Accessing Content

Appendix B

Exercise Repository

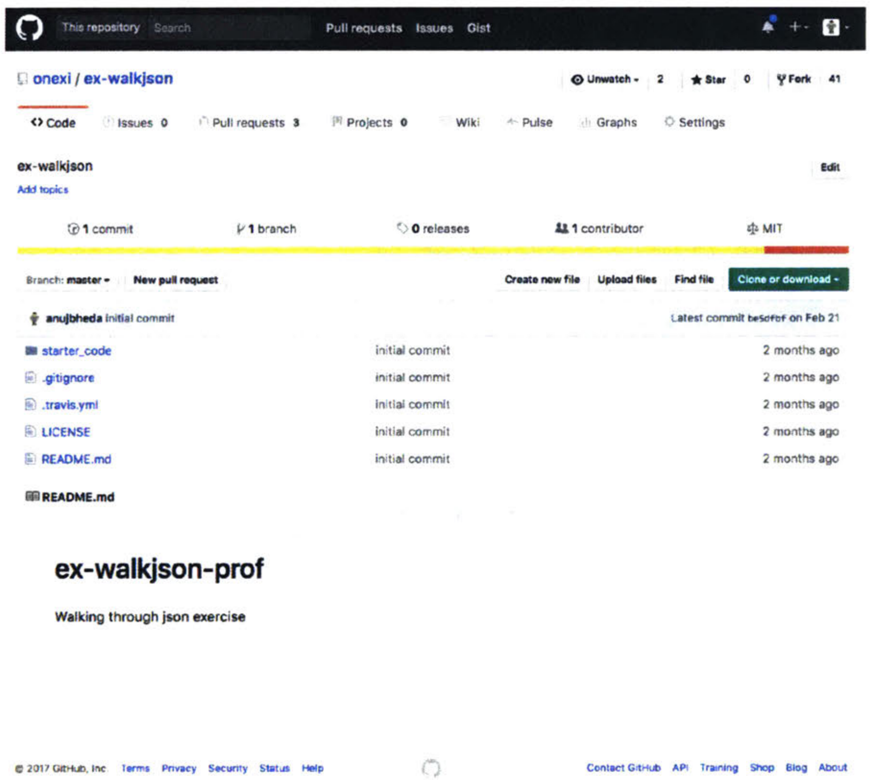


Figure B-1: Exercise Repo on GitHub

Appendix C

IDE with Code Linter

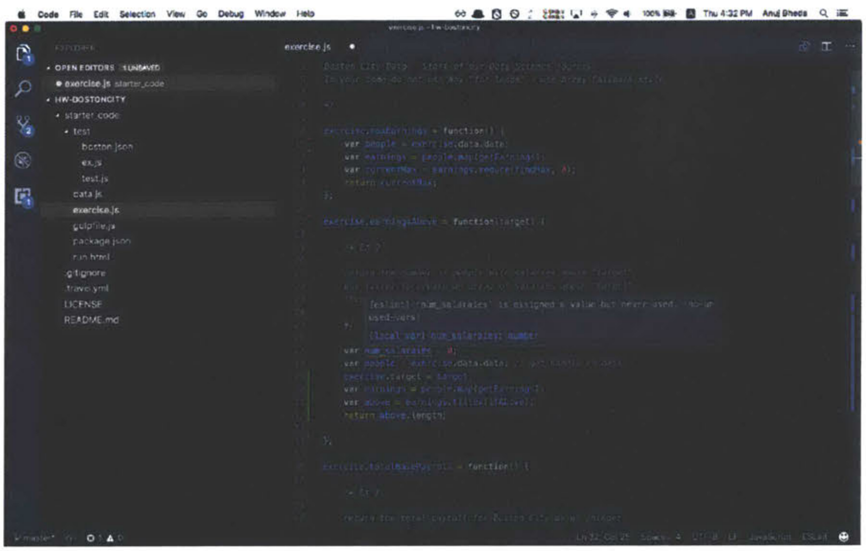


Figure C-1: Visual Studio with ESLint

Appendix D

Test Driven Development

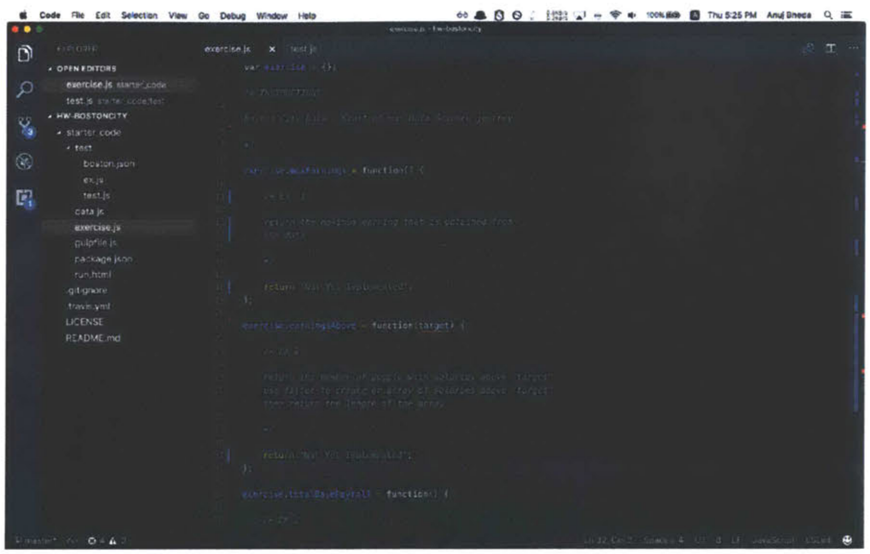


Figure D-1: Placeholder Exercises

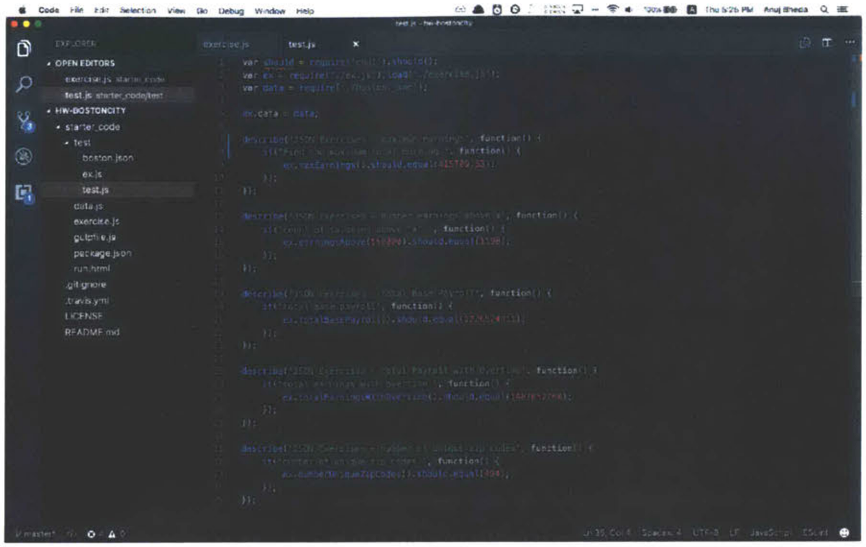


Figure D-2: Unit Tests

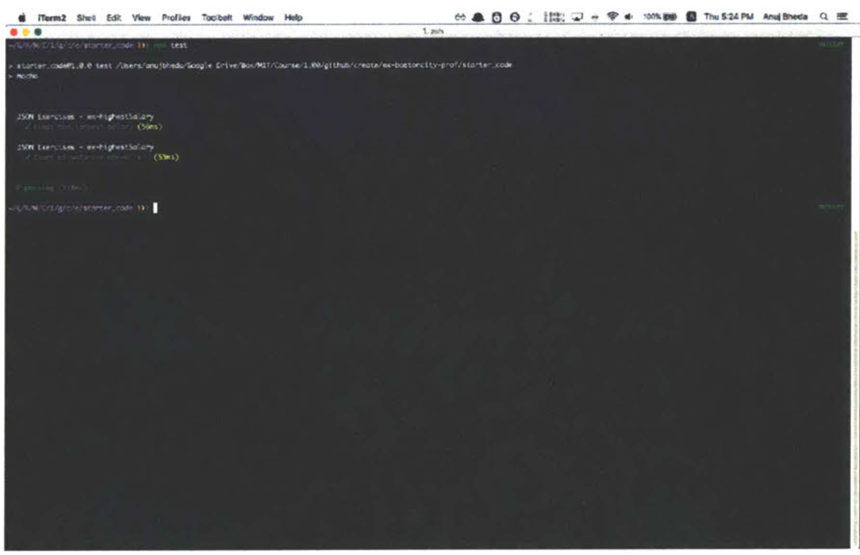


Figure D-3: Results Feedback

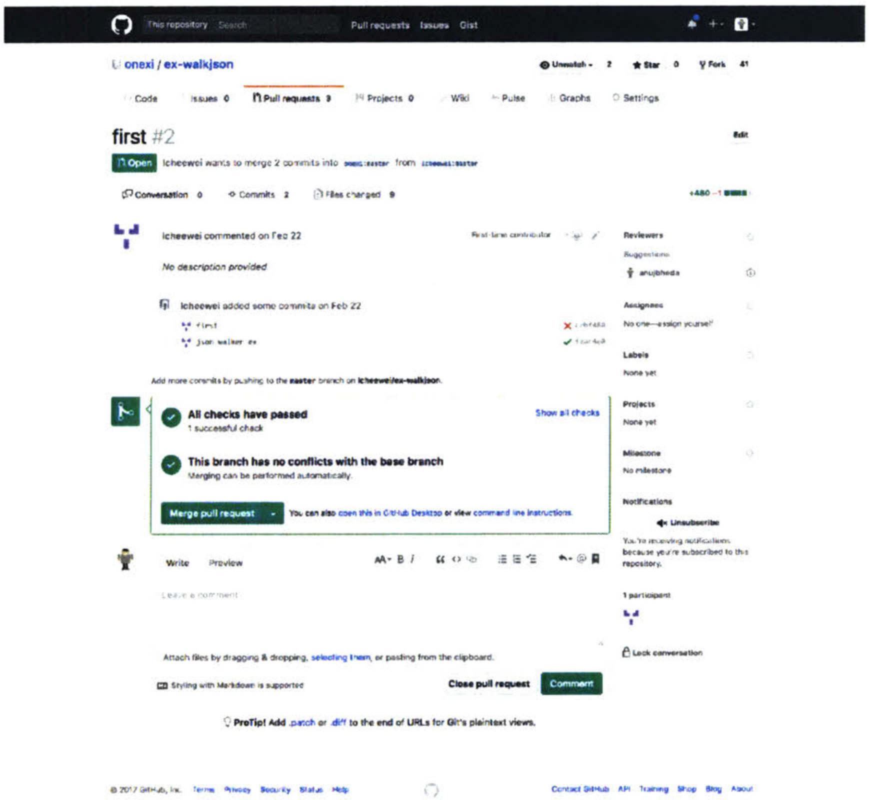


Figure D-4: GitHub Feedback through TravisCI

Appendix E

Piazza

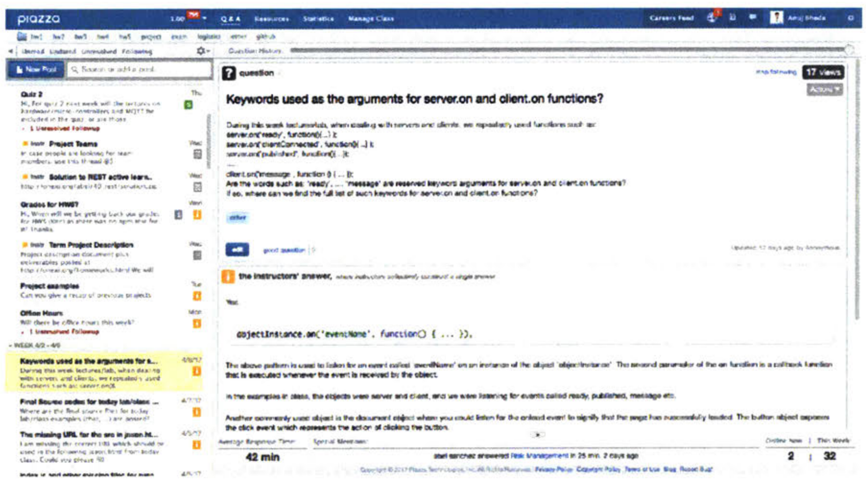
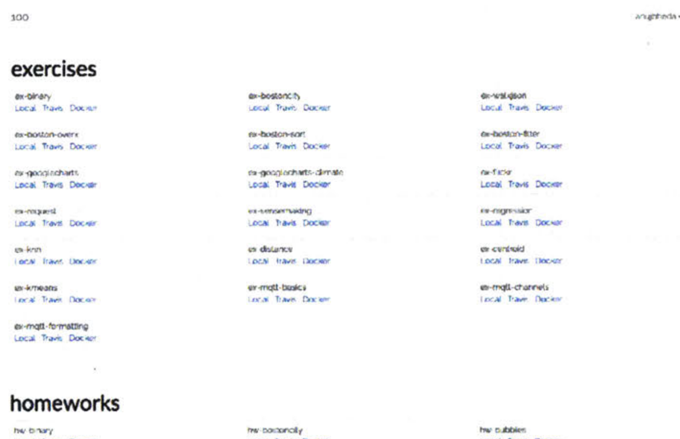


Figure E-1: Piazza

Appendix F

Dashboard



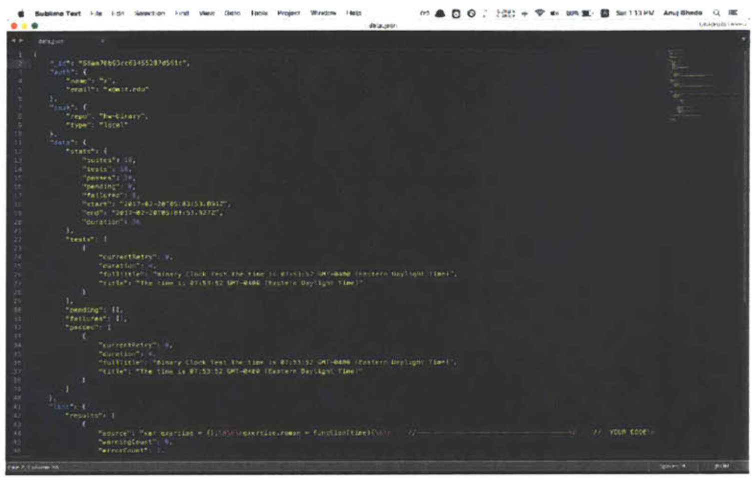
(a) Global Snapshot

(b) Individual Assignment

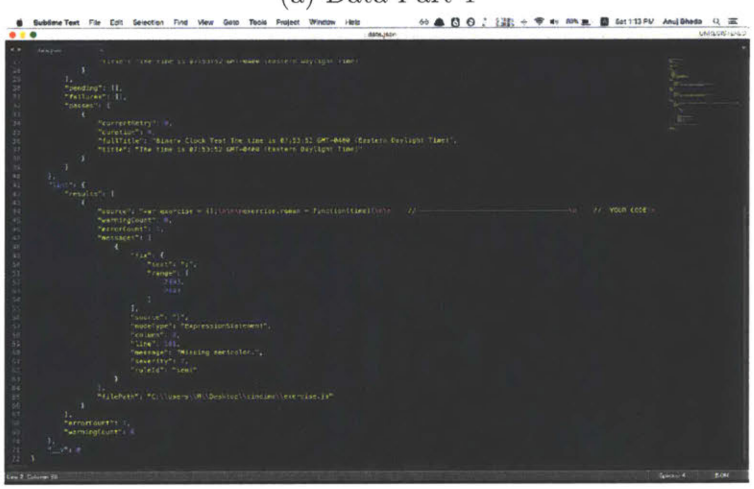
Figure F-1: Dashboard Views

Appendix G

Data



(a) Data Part 1



(b) Data Part 2

Figure G-1: Data Structure

Appendix H

Student Demographics

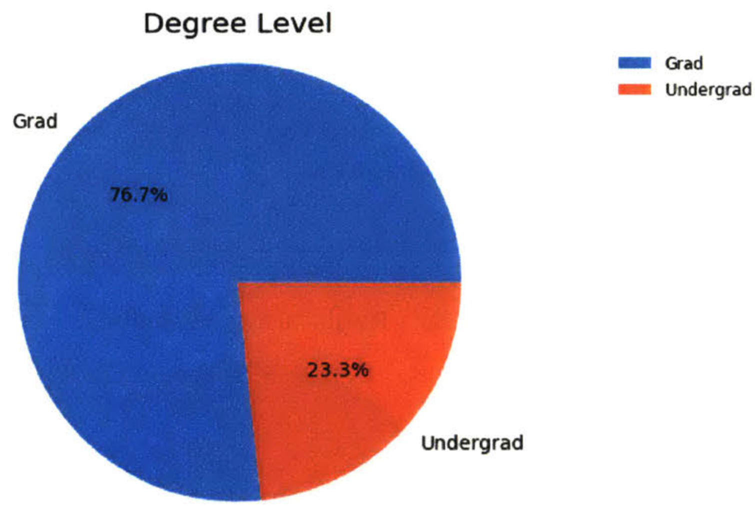


Figure H-1: Degree Level

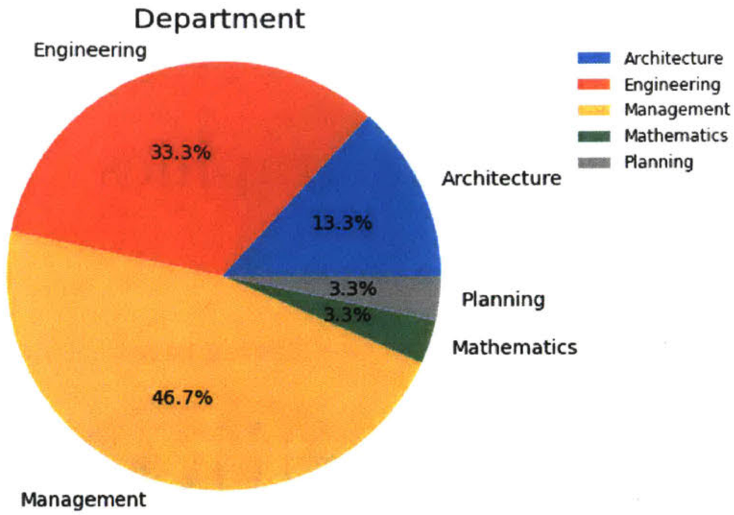


Figure H-2: Departments

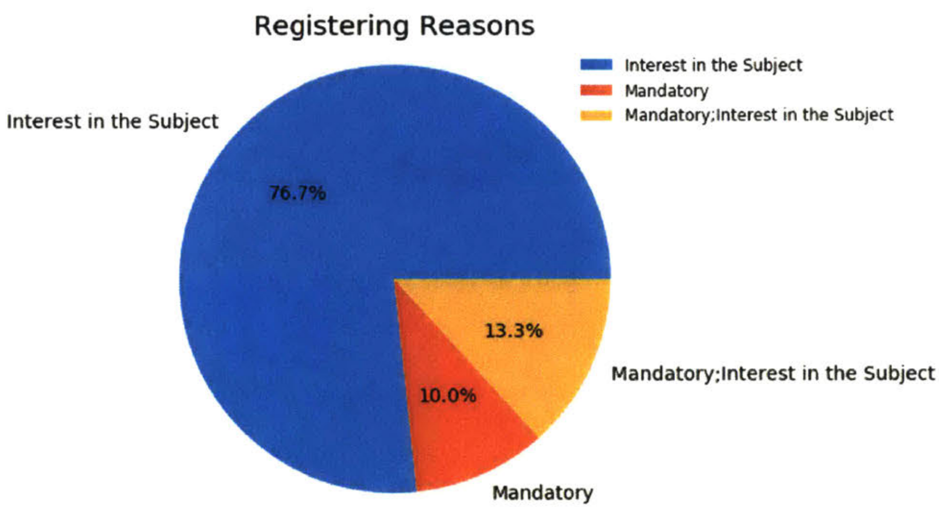


Figure H-3: Registering Reasons

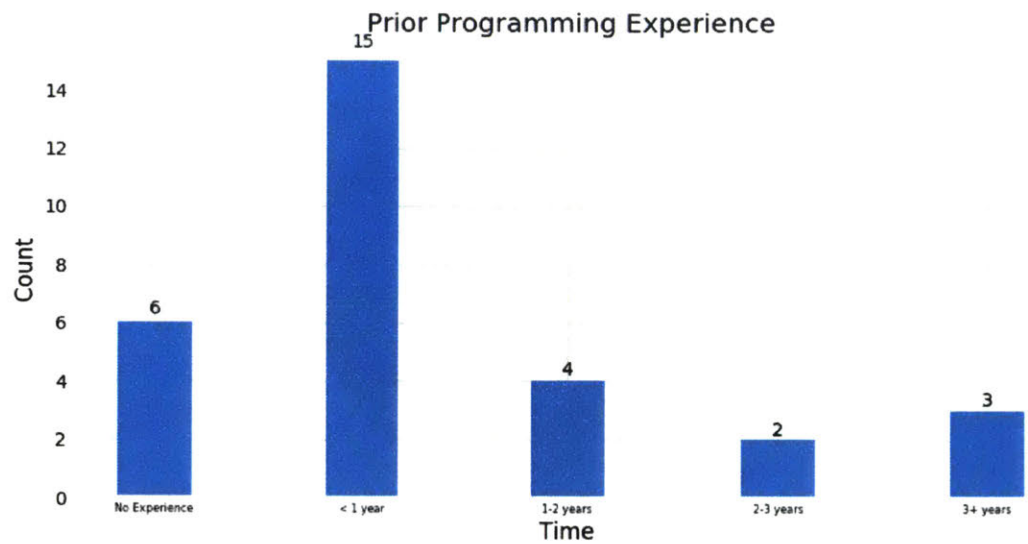


Figure H-4: Prior Programming Experience

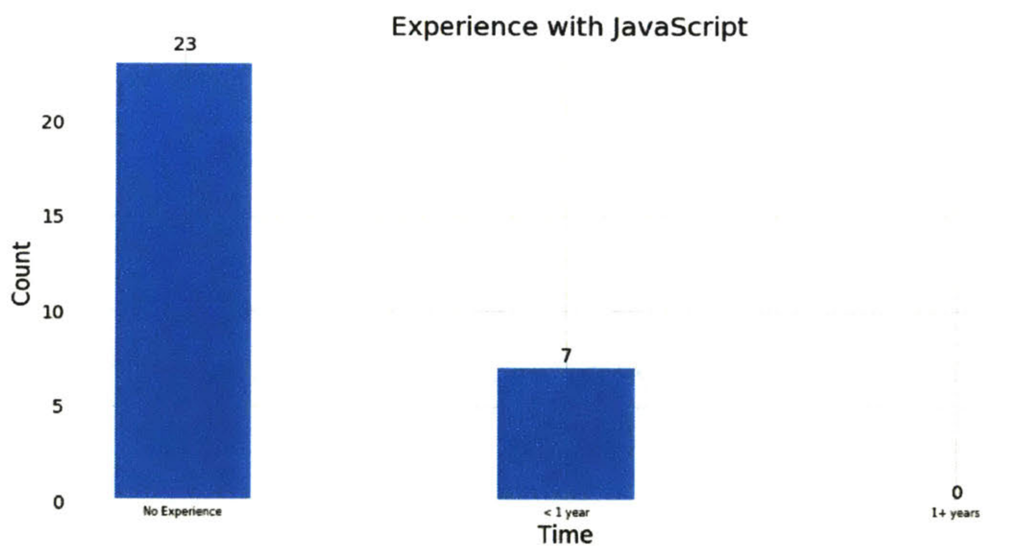


Figure H-5: Experience with JavaScript

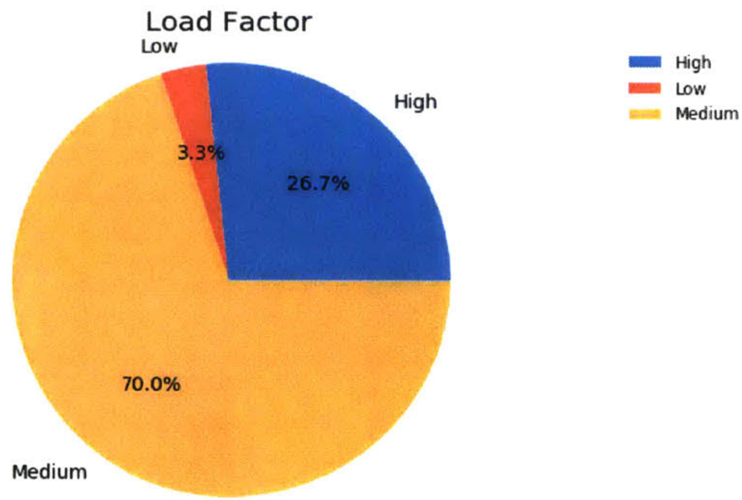


Figure H-6: Load Factor

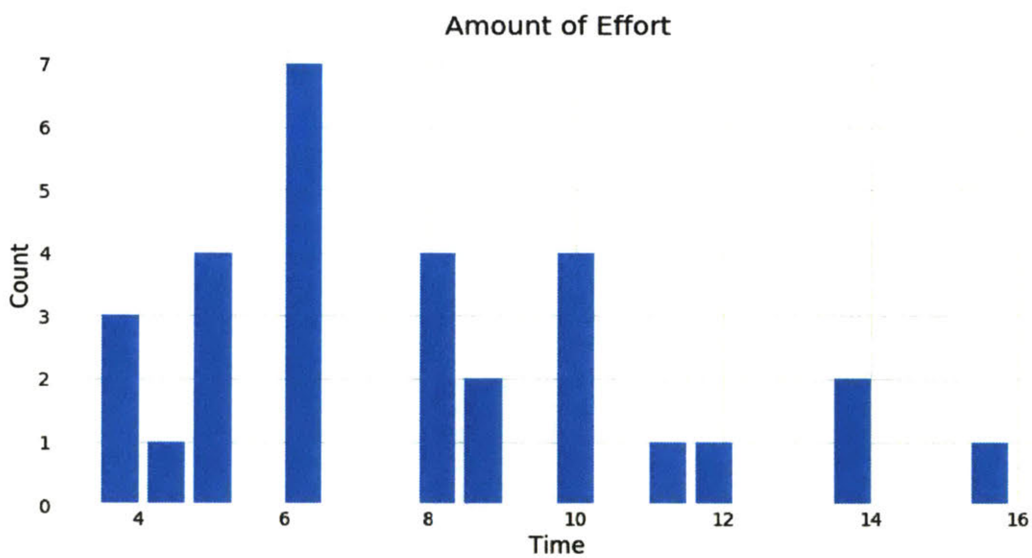
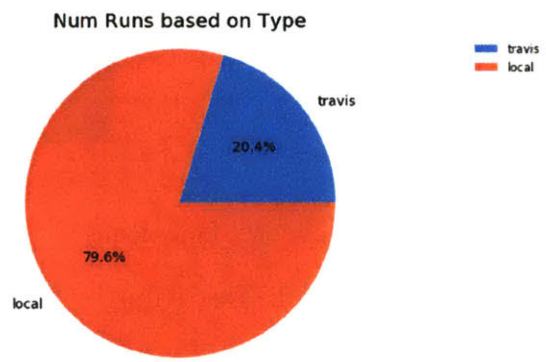


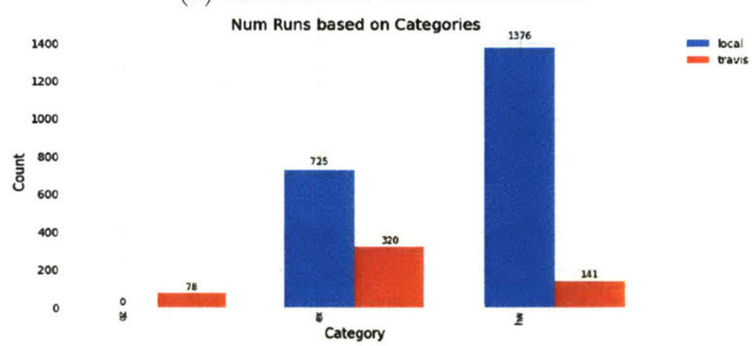
Figure H-7: Amount of Effort

Appendix I

Preliminary Analysis

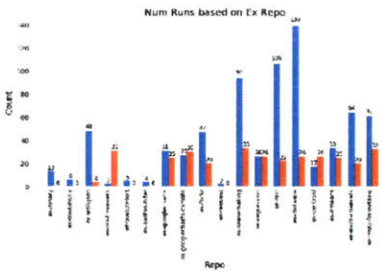


(a) Runs based on Environment

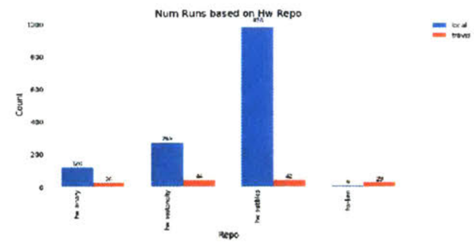


(b) Runs based on Categories

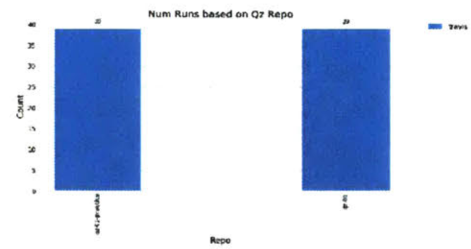
Figure I-1: Number of Test Runs



(a) Exercise Runs



(b) Homework Runs



(c) Quiz Runs

Figure I-2: Test Runs by Category

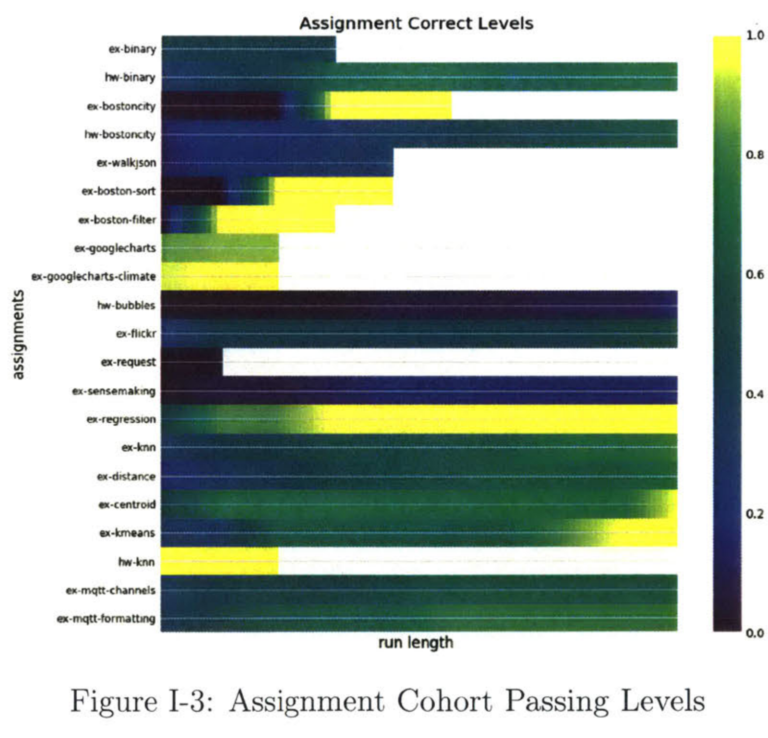


Figure I-3: Assignment Cohort Passing Levels

Appendix J

Neural Network

1. **Feed Forward**

input activation

$$A^0 = \begin{bmatrix} a_1^(0) \\ a_2^(0) \end{bmatrix} = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} \tag{J.1}$$

weights between input and hidden layer

$$W^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{bmatrix} \tag{J.2}$$

hidden layer aggregation

$$Z^{(1)} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(1)} a_1^(0) + w_{12}^{(1)} a_2^(0) \\ w_{21}^{(1)} a_1^(0) + w_{22}^{(1)} a_2^(0) \\ w_{31}^{(1)} a_1^(0) + w_{32}^{(1)} a_2^(0) \end{bmatrix} = (W^1)^T . A^(0) \tag{J.3}$$

hidden layer activation

$$A^{(1)} = \begin{bmatrix} a_1^{(1)} \\ a_2^{(1)} \\ a_3^{(1)} \end{bmatrix} = \begin{bmatrix} \max(0, z_1^{(1)}) \\ \max(0, z_2^{(1)}) \\ \max(0, z_3^{(1)}) \end{bmatrix} = ReLU(Z^{(1)}) \tag{J.4}$$

weights between hidden and output layer

$$W^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(2)} \\ w_{13}^{(2)} & w_{23}^{(2)} \end{bmatrix} \tag{J.5}$$

output layer aggregation

$$Z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} = (W^2)^T.A^{(1)} \tag{J.6}$$

output layer activation

$$A^{(2)} = \begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} = \begin{bmatrix} \frac{e^{z_1^{(2)}}}{\sum_{i=1}^{3} e^{z_i^{(2)}}} \\ \frac{e^{z_2^{(2)}}}{\sum_{i=1}^{3} e^{z_i^{(2)}}} \\ \frac{e^{z_3^{(2)}}}{\sum_{i=1}^{3} e^{z_i^{(2)}}} \end{bmatrix} = softmax(Z^{(2)}) \tag{J.7}$$

defining an one-hot vector

$$T = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \tag{J.8}$$

where $t_i = 1$ if $i = y^{(1)}$

cross entropy loss

$$L = -\sum_{i=1}^{2} t_i \log(a_i^{(2)}) \tag{J.9}$$

2. **Backward Propagation**

weight updates

$$W_{t+1}^{(1)} = W_t^{(1)} - \eta \frac{\partial L}{\partial W_t^{(1)}} \tag{J.10}$$

$$W_{t+1}^{(2)} = W_t^{(2)} - \eta \frac{\partial L}{\partial W_t^{(2)}} \tag{J.11}$$

partial derivative for $W_t^{(2)}$

$$\frac{\partial L}{\partial W_t^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}\frac{\partial Z^{(2)}}{\partial W_t^{(2)}} \tag{J.12}$$

$$\frac{\partial Z^{(2)}}{\partial W_t^{(2)}} = A^{(1)} \tag{J.13}$$

$$\frac{\partial L}{\partial Z^{(2)}} = \delta^{(2)} = A^{(2)} - T \tag{J.14}$$

$$\frac{\partial L}{\partial W_t^{(2)}} = A^{(1)}(A^{(2)} - T) \tag{J.15}$$

partial derivative for $W_t^{(1)}$

$$\frac{\partial L}{\partial W_t^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}\frac{\partial Z^{(1)}}{\partial W_t^{(1)}} \tag{J.16}$$

$$\frac{\partial Z^{(1)}}{\partial W_t^{(1)}} = A^0 \tag{J.17}$$

$$\frac{\partial L}{\partial Z^{(1)}} = \delta^{(1)} = \frac{\partial L}{\partial Z^{(2)}}\frac{\partial Z^{(2)}}{\partial Z^{(1)}} = \delta^{(2)}\frac{\partial z^{(2)}}{\partial Z^{(1)}} \tag{J.18}$$

$$\frac{\partial Z^{(2)}}{\partial Z^{(1)}} = Diag[Z^{(1)}(1 - Z^{(1)})]W^{(2)} \tag{J.19}$$

$$\frac{\partial L}{\partial Z^{(1)}} = Diag[Z^{(1)}(1 - Z^{(1)})]W^{(2)}\delta^{(2)} \tag{J.20}$$

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Kimberly E. Arnold and Matthew D. Pistilli. Course signals at purdue: Using learning analytics to increase student success. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*, LAK '12, pages 267–270, New York, NY, USA, 2012. ACM.

[2] Yoshua Bengio, Patrice Y Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 1994.

[3] M Berland, T Martin, and T Benton. Using learning analytics to understand the learning pathways of novice programmers. *Journal of the*, 2013.

[4] Paulo Blikstein. *Using learning analytics to assess students' behavior in open-ended programming tasks.* ACM, New York, New York, USA, February 2011.

[5] Mohamed Amine Chatti, Anna Lea Dyckhoff, Ulrik Schroeder, and Hendrik Thüs. A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, February 2013.

[6] Brenda Cheang, Andy Kurnia, Andrew Lim, and Wee-Chong Oon. On automated grading of programming assignments in an academic institution. *Computers & Education*, 41(2):121–131, September 2003.

[7] Doug Clow. An overview of learning analytics. *Teaching in Higher Education*, September 2013.

[8] Jeremiah E Dittmar. Information Technology and Economic Change: The Impact of The Printing Press. *The Quarterly Journal of Economics*, 126(3):1133–1172, August 2011.

[9] Rebecca Ferguson. Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, February 2013.

[10] Chan Yuen Fook and Gurnam Kaur Sidhu. Investigating learning challenges faced by students in higher education. *Procedia - Social and Behavioral Sciences*, 186:604 – 612, 2015.

[11] Charles R Graham. Blended learning systems. *Handbook of blended learning: Global perspectives, local designs*, pages 3–21, 2006.

[12] Rhys Grossman. The Industries That Are Being Disrupted the Most by Digital. *Harvard Business Review*, March 2016.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[14] Jennifer Hofmann. Why blended learning hasnâĂŹt (yet) fulfilled its promises. *Handbook of blended learning: Global perspectives, local designs*, pages 27–40, 2006.

[15] Holger M Kienle. It's About Time to Take JavaScript (More) Seriously. *IEEE Software*, 27(3):60–62, 2010.

[16] Nancy J Lewis and Peter Z Orton. Blending learning for business impact. *Handbook of blended learning: Global perspectives, local designs*, page 61, 2006.

[17] Amrien Hamila Ma'arop and Mohamed Amin Embi. Implementation of blended learning in higher learning institutions: A review of the literature. *International Education Studies*, 9(3):41 – 52, 2016.

[18] T. Mahalingam and A. V. Rajan. Cloud and mobile computing: Affordances of the 21st century teaching and learning. In *2013 International Conference on Current Trends in Information Technology (CTIT)*, pages 125–128, Dec 2013.

[19] J. Ng, D. Ruta, A. A. Rubaie, D. Wang, L. Powell, B. Hirsch, L. Ming, C. Ling, and A. A. Dhanhani. Smart learning for the next generation education environment. In *2014 International Conference on Intelligent Environments*, pages 333–340, June 2014.

[20] S Nunn, J T Avella, and T Kanai. Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online ...*, 2016.

[21] Julieth E. Ospina-Delgado, Ana Zorio-Grima, and MarÃŋa A. GarcÃŋa-Benau. Massive open online courses in higher education: A data analysis of the mooc supply. *Intangible Capital*, 12(5):1401 – 1450, 2016.

[22] Roger S Pressman. *Software engineering: a practitioner's approach.* Palgrave Macmillan, 2005.

[23] Shaymaa E Sorour, Tsunenori Mine, Kazumasa Goda, and Sachio Hirokawa. Predicting students' grades based on free style comments data by artificial neural network. In *2014 IEEE Frontiers in Education Conference (FIE)*, pages 1–9. IEEE, 2014.

[24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[25] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *ICML (3)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 1139–1147. JMLR.org, 2013.

[26] J Voogt, O Erstad, C Dede, and P Mishra. Challenges to learning and schooling in the digital networked world of the 21st century. *Journal of Computer Assisted Learning*, 29(5):403–413, October 2013.

[27] Lisa Wang, Angela Sy, Larry Liu, and Chris Piech. Deep knowledge tracing on programming exercises. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pages 201–204, New York, NY, USA, 2017. ACM.

[28] Christopher Watson, Frederick W B Li, and Jamie L Godwin. Predicting Performance in an Introductory Programming Course by Logging and Analyzing Student Programming Behavior. In *2013 IEEE 13th International Conference on Advanced Learning Technologies (ICALT)*, pages 319–323. IEEE, 2013.

[29] Zamzami Zainuddin and Siti Hajar Halili. Flipped classroom research and trends from different fields of study. *The International Review of Research in Open and Distributed Learning*, 17(3), 2016.