A black rectangle with a black background

Description automatically generated with low confidenceLogo, company name

Description automatically generated

DAEN 690

Project Report

Darvik Kunal Banda

Erick A Torres

Joe B Brock

Sagar D Goswami

Telina Andrianaivo

Spring 2023

Predicting CEC Graduate Course Demand

**About the Cover**

Dr. Isaac Gang is an Associate Professor at the George Mason University College of Engineering and Computing, Volgenau School of Engineering, MS Data Analytics Engineering (DAEN) program.

He joined the DAEN faculty in the Fall of 2020 from Texas A&M University-Commerce (TAMUC) where he served as an Assistant Professor of Computer Science as well as the department’s Outreach Coordinator. Before coming to TAMUC, Dr. Gang was an Assistant Professor of Computer Science and Engineering at the University of Mary Hardin-Baylor (UMHB) and an Adjunct Professor of Computer Science at the University of Southern Mississippi’s School of Computing before joining UMHB.

Dr. Gang is a former DOE grant winner, former President and Board Member of the Association of Computer Educators in Texas (ACET), Industry Advisory Board (IAB) Coordinator, and the Director of CS For All.

His current and primary teaching responsibilities at Mason largely involves Data Analytics Engineering graduate courses along with a mix of CS and AIT graduate courses. He is an affiliate faculty member of GMU’s C4I & Cyber Center.

Dr. Gang’s primary research agenda involves Big Data Analytics (emphasis on data bias and data governance), Cyber Security (ransomware, steganography, and cyberbullying), and Image/Signal Processing.

Contents

Table of Contents

[Abstract 5](#_Toc127138363)

[Section 1: Problem Definition 7](#_Toc127138364)

[1.1 Background 7](#_Toc127138365)

[1.2 Problem Space 11](#_Toc127138366)

[1.3 Research 12](#_Toc127138367)

[1.3.1 Enrollment Predictions with Machine Learning 12](#_Toc127138368)

[1.3.2 Machine Learning Methods for Course Enrollment Prediction 13](#_Toc127138369)

[1.3.3 Music Recommendation System 13](#_Toc127138370)

[1.4 Solution Space 14](#_Toc127138371)

[1.5 Project Objectives 15](#_Toc127138372)

[1.6 Primary User Stories 16](#_Toc127138373)

[1.7 Product Vision 16](#_Toc127138374)

[1.7.1 Scenario #1 16](#_Toc127138375)

[1.7.2 Scenario #2 16](#_Toc127138376)

[Section 2: Datasets 16](#_Toc127138377)

[2.1 Overview 16](#_Toc127138378)

[2.2 Field Descriptions 16](#_Toc127138379)

[2.3 Data Context 17](#_Toc127138380)

[2.4 Data Conditioning 18](#_Toc127138381)

[2.5 Data Quality Assessment 18](#_Toc127138382)

[2.6 Other Data Sources 18](#_Toc127138383)

[2.7 Storage Medium 18](#_Toc127138384)

[2.8 Storage Security 18](#_Toc127138385)

[2.9 Storage Costs 19](#_Toc127138386)

[Section 3: Algorithms & Analysis / ML Model Exploration & Selection 19](#_Toc127138387)

[3.1 Solution Approach 19](#_Toc127138388)

[3.1.1 Systems Architecture 19](#_Toc127138389)

[3.1.2 Systems Security 19](#_Toc127138390)

[3.1.3 Systems Data Flows 19](#_Toc127138391)

[3.1.4 Algorithms & Analysis 19](#_Toc127138392)

[3.2 Machine Learning 19](#_Toc127138393)

[3.2.1 Model Exploration 20](#_Toc127138394)

[3.2.2 Model Selection 20](#_Toc127138395)

[Section 4: Visualizations / ML Model Training, Evaluation, & Validation 20](#_Toc127138396)

[4.1 Overview 20](#_Toc127138397)

[4.2 Visualizations 20](#_Toc127138398)

[4.3 Machine Learning 20](#_Toc127138399)

[4.3.1 Model Training 20](#_Toc127138400)

[4.3.2 Model Evaluation 21](#_Toc127138401)

[4.3.3 Model Validation 21](#_Toc127138402)

[Section 5: Findings 21](#_Toc127138403)

[Section 6: Summary 21](#_Toc127138404)

[Section 7: Future Work 21](#_Toc127138405)

[Appendix A: Glossary 23](#_Toc127138406)

[Appendix B: GitHub Repository 24](#_Toc127138407)

[Overview 24](#_Toc127138408)

[GitHub Repository Link 24](#_Toc127138409)

[GitHub Repository Contents 24](#_Toc127138410)

[Appendix C: Risks 25](#_Toc127138411)

[Sprint 1 Risks 25](#_Toc127138412)

[Sprint 2 Risks 25](#_Toc127138413)

[Sprint 3 Risks 25](#_Toc127138414)

[Sprint 4 Risks 26](#_Toc127138415)

[Sprint 5 Risks 26](#_Toc127138416)

[Appendix D: Agile Development 27](#_Toc127138417)

[Scrum Methodology 27](#_Toc127138418)

[Sprint 1 Analysis 28](#_Toc127138419)

[Sprint 2 Analyis 29](#_Toc127138420)

[Sprint 3 Analysis 29](#_Toc127138421)

[Sprint 4 Analysis 29](#_Toc127138422)

[Sprint 5 Analysis 29](#_Toc127138423)

[Works Cited 31](#_Toc127138424)

[Figure 1 TIme to Graduate - George Mason University (Data USA) 7](#_Toc127138425)

[Figure 2 GMU Full-Time to Part-Time Student Ratio (Data USA) 8](#_Toc127138426)

[Figure 3 Harvard University - Time to Complete (Data USA) 9](#_Toc127138427)

[Figure 4 Time to Complete – JMU (Data USA) 9](#_Toc127138428)

[Figure 5 SAT Scores – JMU (Data USA) 10](#_Toc127138429)

[Figure 6 SAT Scores – GMU (Data USA) 10](#_Toc127138430)

[Figure 7 Comparing ML Methods for Course Enrollment Prediction (Shao, Ieong, Levine, Stronach, & Fan, 2022) 13](#_Toc127138431)

[Figure 8 Simplified Cloud Architecture for the Project 15](#_Toc127138432)

[Figure 9 Sprint 1: Scrum Planning Board 27](#_Toc127138433)

[Figure 10 Sprint Planning Sheet 28](#_Toc127138434)

[Figure 11 Sprint 1: Agile Board 28](#_Toc127138435)

**This page intentionally left blank**

Abstract

Abstract

**INSTRUCTIONS**

[NOTE: The project abstract is a separately graded assignment in the course. The final approved project abstract is to be copied word-for-word from the other assignment into this report.]

Write one paragraph of no more than 300 words that summarizes your project. Here are the typical kinds of information found in most abstracts which you should use as an outline as you develop your abstract.

1. The context or background information for your research; the general topic under study; the specific topic of your research.
2. The central questions or statement of the problem your research addresses.
3. What’s already known about this question, what previous research was conducted or shown.
4. The main reason(s), the exigency, the rationale, the goals for your research — why is it important to address these questions? Are you, for example, examining a new topic? Why is that topic worth examining? Are you filling a gap in previous research? Applying new methods to take a fresh look at existing ideas or data? Resolving a dispute within the literature in your field?
5. Your research and/or analytical methods.
6. Your main findings, results, or arguments.
7. The significance or implications of your findings or arguments.

Your abstract should be intelligible on its own, without a reader’s having to read your entire paper.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

**This page intentionally left blank**

Report

# Problem Definition

## Background

George Mason University (GMU) has grown due to its high acceptance rate (89.2%), high retention rate (86%), and low tuition cost (Data USA). Yet, waitlisting remains a consistent issue for the Office of Provost. Waitlisted students who cannot apply for their required classes are stuck. They won't be able to apply until a slot opens for their current semester. Waiting another semester could entail an extended student graduation date. This is a costly problem, as longer graduation dates mean smaller employment prospects. Graduating outside of the typical window is unattractive to employers. Consider that 32% of all Mason students are part-time graduates. A sizeable part of Mason's community is in danger of losing potential job prospects because of waitlisting.

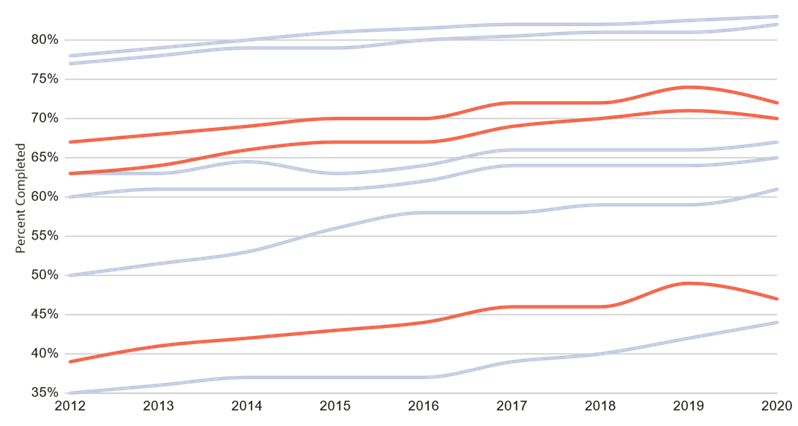


Figure TIme to Graduate - George Mason University (Data USA)

Waitlisting also creates volatile class sizes for educators. Large classes encourage professors to forgo individual student needs for collective needs. Students may have to engage with stressed professors doing crowd control. It creates an isolating environment for the student and professor. Students cannot receive the help they need, while professors cannot meet student needs. When students mass-drop a class, the results are more harmful. There are governmental grants to keep students in certain classes. The school does not receive those benefits if too many students drop a course. These problems may be outside of the school's control. Student interest contributes to whether they will stay in a class. Unpredictability among graduate students is not uncommon, but reducing it is a possibility. It could provide future funding for GMU through government grants. It would reduce student issues and assist those who are the most affected by waitlisting.

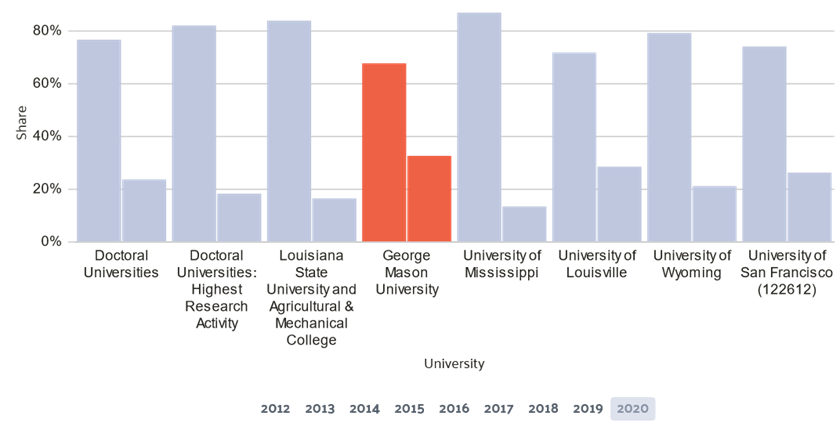


Figure GMU Full-Time to Part-Time Student Ratio (Data USA)

This includes international students who migrate to the United States for better opportunities. These students must have at least 12 credits per semester for their F1 visa to be legitimate. If international students cannot pay for their classes, they can't keep their visas. Waitlisting exasperates this problem by creating competition among international students. Being unable to register for classes places these students in a high stress scenario. It is already difficult to get these F1 visas accepted. GMU damages its reputation when international students cannot get support for their visas. Waitlisting is an oversight that needs correction.

Online courses can ease waitlisting as there's less physical space needed. This is a good step forward, but online courses are not accessible to all students. International students can only take one online course per semester. It may exasperate professor-student communications because it's harder to work through problems online. This doesn't completely reduce waitlisting either. It only provides a more convenient set of options for students on how they want to manage their time. The 47% of students graduating on time may appreciate online classes. The other 63% will hold negative opinions about their education at GMU. This cost threatens GMU’s student satisfaction, funding, time, and reputation. Consider Harvard’s “Time to Complete”:

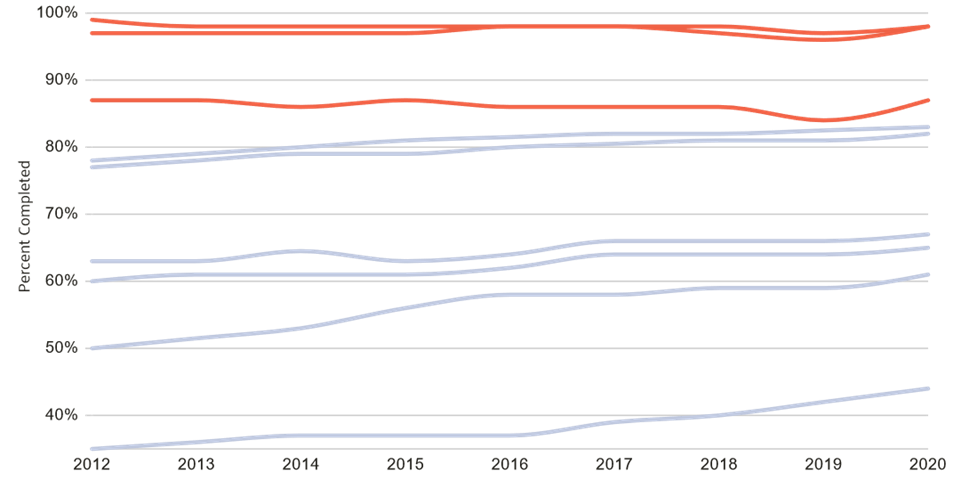


Figure Harvard University - Time to Complete (Data USA)

Harvard is one of the top schools in the United States, partly because students graduate on time. It’s not the only example. James Madison University (JMU) is closer to GMU via its university ranking. Both universities have similar net prices ($18,592 JMU, $18,285 GMU). Their student expenses ($10,938 JMU, $12,105 GMU) and loan default rates (2.05% JMU, 2.58% GMU) are only somewhat different. JMU is not much better than GMU. Where JMU succeeds is in its "time to complete". JMU applicants could enjoy smaller net costs because more students graduate on time. Compare this to GMU’s cost growth within one year (9.29%).

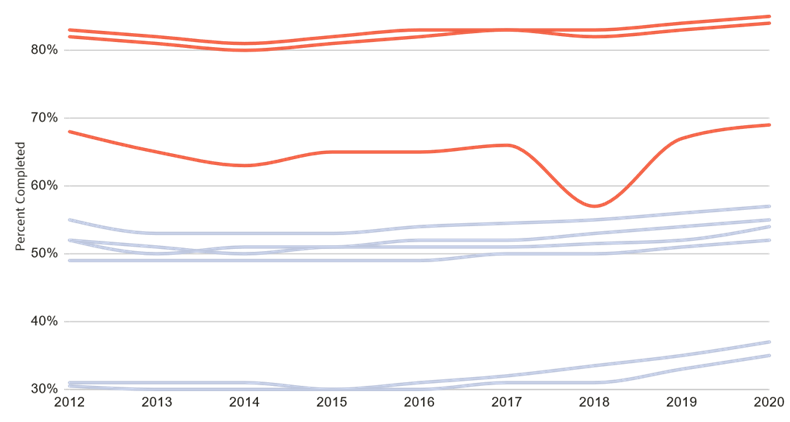


Figure Time to Complete – JMU (Data USA)

This could be because JMU does not have to deal with so many part-time students. Most of JMU's student population is homogeneous, with different educational goals then GMU.



Figure SAT Scores – JMU (Data USA)

JMU is a good school, but their SAT requirements are not as good as GMU. It does prevent JMU from ranking higher than GMU. Despite this, GMU suffers from gaining a higher ranking because of it's time to graduate.

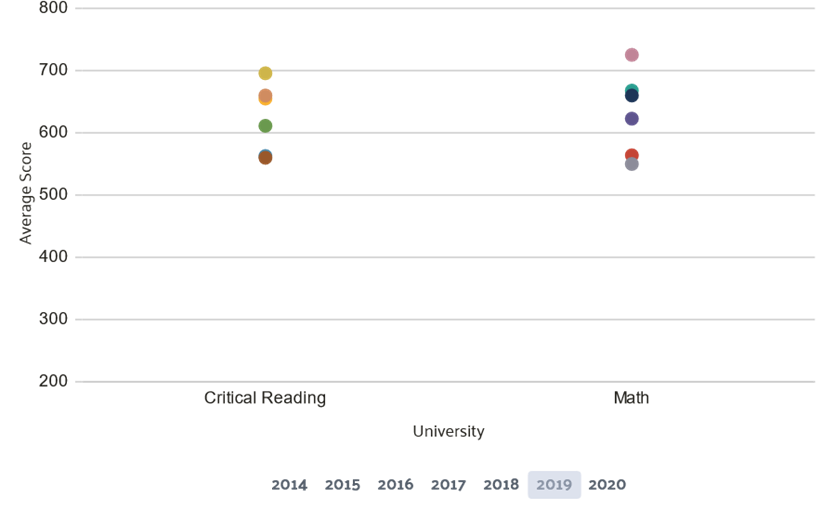


Figure SAT Scores – GMU (Data USA)

George Mason University is attractive, but its late graduation times harm student appeal. The Office of Provost identified four primary areas of focus for this project. These include Core Courses, Electives, Pre-Requisites classes, and Study Interests. These are key issues to predict if a class needs waitlisting. This is helpful because it can identify patterns. Students are subject to use different strategies for obtaining their desired class schedule. One tactic students will use include applying for many waitlisted classes to receive one they need. Identifying what strategies students use can identify which classes are prone to waitlisting. Recognizing complexity is also important. This problem becomes egregious when accounting for grade levels, funding, and drop penalties to classes. The COVID pandemic has only worsened these issues and introduced potential temporal bias into our data set. This project will explain in detail the solutions installed to combat these problems. Each section presents facets for the ideation, implementation, and resolution of waitlisting problems. This is not a completed list. It's possible to tool more solutions for predicted and minimizing waitlisted students. This project’s purpose is to provide a generalized solution to finding the optimal number of students.

## Problem Space

Predicting waitlisted seats is a pressing concern for the Enrollment Management Services at George Mason University, but it is not the only factor. The primary motive for this project is to reduce the wastage of monetary resources due to over-booked/under-booked class sessions. Minimizing waitlisted seats is an avenue to do so, but it is not relevant for all classes. Having students waitlisted may be acceptable. Electives and non-STEM courses fall under this umbrella. Constraints to this project will inform its outcome, so complexity reduction is key. Reducing complexity begins by analyzing the number of Pre-Requisite classes. Pre-Requisites hold precedence above any other classes. This is because Pre-Requisites prevents students from taking other classes. Students who can't complete a Pre-Requisite to another class have to wait. If the class is competitive, it will encourage a flooded waitlist. What is important is measuring the "depth" of Pre-Requisites for each class. Finding classes which create bottlenecks for students should clarify to organizers difficult classes. It also assists in measuring Student Interests.

Student Interest is volatile because students have different goals and aims. It is also a focus in our project's consolidation effort. There are only a few ways to measure Student Interest. Student drop rate, student withdrawals, and degree changes all contribute to Student Interest. Yet, interest is very fluid. Students can make these changes for any reason, but the qualifiers listed make for an excellent control against waitlisting. Core classes and Electives are the least susceptible to change. Core classes can experience a similar pattern of behavior to Pre-Requisites. It is less likely, but if a degree is competitive enough students may fill up a class. It is a priority for student to complete Core Classes, but it may be dependent on their program. Some students are part of a cohort that only takes on class per semester. It may be acceptable for them to have a longer graduation window as a result. Electives are similar, but what drives these two is when they are important. Electives are more important to account for at the end of the year. Core Classes are crucial during the beginning. Both can be Pre-Requisites for other classes, so it provides a purview to what our focus should be.

Non-waitlisted courses, Bridge Courses, Labs, and Scheduling Conflicts are special cases. These come about because some classes have specific requirements to begin. Bridge Courses, for example, force potential graduate students to enroll in undergraduate classes to start their master’s program. This constraint filters students based on their academic eligibility. It also could predict their performance for a specified degree. Engineering students may have the same classes as Computer Scientists but may fail more classes on average. This could be because they don’t have the same professional knowledge. Labs are a similar problem constraint because they need to operate in conjunction with a given class. If a class is available, but the lab is not, the student might not be able to complete a necessary part of their degree. Different specialized constraints must similarly handle the data aspect of our classes.

Waitlist-enabled classes have a maximum capacity of 99 by default. This raises expectations for the total class size, since waitlisting operates on a first come/first serve basis. Specializations can exasperate this problem, as more competitive fields have fewer open spots available. Whether the class is online affects the student ability to graduate on time. International Students are bounded by their visa status, and if their required classes are online only it may cause issues in the future. Student’s scheduling conflicts can inflame waitlisting problems. Classes that are assigned for hours will dissuade certain majors from applying to them. It may be difficult for professors to coordinate with others on which time slots are available. More timeslots would be ideal, but there is a limited number of professors to assign those timeslots.

Working around these issues is an optimization problem which requires time and scheduling. All problems listed are constraints. The primary fear is not leaving the project incomplete, but finding a resolution that is ill-fated for minimizing waitlisting positions. Most post-COVID data will likely have to be scrapped or thoroughly examined. This includes the initial starting months of the pandemic and the months thereafter. Student sentiment and purchasing decisions during these months will have to be heavily monitored, as they indicate what changed throughout this time period if the COVID-era is going to be used. Pre-Requisite course selection is the most important field, and aggregation functions are needed to count how many pre-requisites are required for each class. Labs must be fitted with any pre-requisite and core classes that are found. Core Course’s size must be identified and compared against Pre-Requisite courses. The problems presented by these solutions are varied. It might not be enough to reduce the bias, and inversely it could remove too much data. The total graduate student population is expected to be low, but focusing on Pre-Requisite courses is the best way to handle the variability.

Students will have to be split into different groups based on the faculty that they have available, the types of courses that they have, and their academic standing. Filtering by online and in-person classes would help create clearer data, but it is dependent on ease-of-use. T-Tip Funding Goals will be a massive constraint for this project. It will prioritize which waitlisted classes that provide the largest bankroll for the university. This could cause a serious collection bias, so it must be monitored.

These provided constraints help guide the project to a unified solution, but other smaller issues are present based on the technology. AWS provides a technical framework for how our solutions will come together within the GMU region. The shortcomings of using AWS will require time, money, and space, provided that GMU decides to expand this service to other AWS regions. The licensing for LP solvers could harm GMU if the technology does not turn out to be as useful. There are many technical failings that could occur, from a lack of technical staff managing the cloud environment to poor configuration management.

Disk Size and CPU usage is likely not going to be an issue for any data scientists implementing the system described by this paper following its completion. What will be an issue is a lack of communication between the computer science side of this equation and the data science/analytics side. The technical knowledge listed throughout this paper must be used to understand and resolve the issues provided. The research provided presents examples of how this was done in prior projects and is used as an inspiration for this project.

## Research

Enrollment Management involves making many critical decisions to ensure the proper functioning of the university. Various data analytical tools and techniques have been used to work on multiple aspects of Enrollment Management. One of the recurrent areas of research in Enrollment Management is making predictions for the course demand, subject to various constraints and previous enrollment data. The most common techniques used to do so include simple temporal projections, multi-variate regression models, Time-Series Models, and, very recently, Deep Learning, among many others. Each technique focuses on different variables/factors and datasets deemed relevant for their case. Most current methods use a lot of data regarding records and the number of external variables affecting the demand. The following ones are most relevant to our project:

### Enrollment Predictions with Machine Learning

Researchers at California State University: Channel Islands (CSU-CI) conducted a research study where they demonstrated a proof of concept of using Data Analytics techniques in AWS SageMaker to predict Student Enrollments at California State University. They targeted this research towards SEM and Data Analytics Technical Practitioners, specifically making Enrollment Management decisions (Soltys, Dang, Reilly, & Soltys, 2021).

They used a combination of various ML Algorithms like AWS SageMaker XGBoost (Extreme Gradient Boosting) and Time-Series Analysis to predict the probability of a student enrolling. The data was distributed in 80% Training Set and 20% Test Set. The frequency distribution of Student Enrollment Probabilities was then filtered by a set threshold value of 0.09 (9%) and summed to obtain the expected student enrollment for that term.

They also remarked that while the predictions are made at the student level, this technique does not provide a definitive answer on whether a student will enroll. However, it performs exceptionally well when aggregating the results to make predictions for the entire batch of students for that term. Even with limited data points, they could predict True-Negatives at 75% accuracy and True-Positives at 61%. By considering more variables, setting weights to penalize False-Negatives, and setting some tolerance values, the University can make better estimates to ensure enough resource allocation for student enrollments well in advance.

### Machine Learning Methods for Course Enrollment Prediction

The researchers at San Diego State University (SDSU) conducted research on predicting course enrollment based on various metrics pertaining to University, Demographics, Admissions, etc. They used previous research on Conditional Probability Analysis as a base model and improved on it using Tree Models like Classification and Regression Trees (CART) and Random Forest (Shao, Ieong, Levine, Stronach, & Fan, 2022).

In the Conditional Probability Analysis technique, the students are divided into various categories like First-Time/Current/Transfer Students, Pre-requisites met/unmet, First Attempt/Multiple attempts, etc. Predictions for the probability of students registering for this course were computed for each of those categories, and summed to get the final student enrollment prediction for a given course.

Additionally, they also worked on CART and Random Forest Models, building on the Conditional Probability. From the results, it was clear that CART performed better than Conditional Probability. However, it was observed that the error for both the models exploded, when predicting 2 years in future. Random Forest on the other hand, gave consistent results with a very low error rate for both the years. The error rates for all these models are as follow:

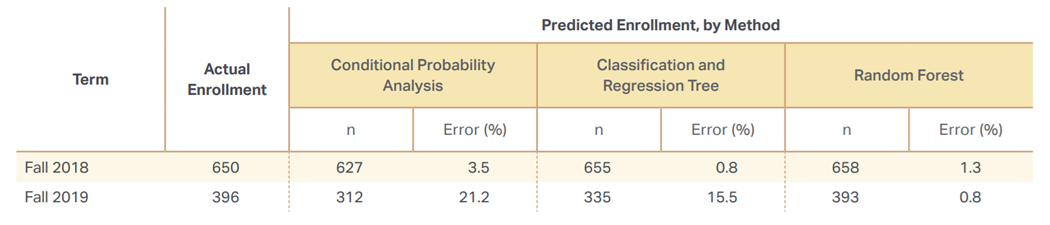


Figure Comparing ML Methods for Course Enrollment Prediction (Shao, Ieong, Levine, Stronach, & Fan, 2022)

### Music Recommendation System

Music Streaming Services have seen a rapid growth in the past decade and are no longer limited to just streaming music on demand, but rather also implementing various ML algorithms, and AI recommendation systems to suggest songs to the users based on their tastes, listening history, and interactions. “A Music Recommendation System with Mathematical Optimization” is one such work, that uses Gurobi Optimizer (Optimization Tool) to create a recommendation system using a mixture of predictive and prescriptive analytics (Gurobi Optimization, 2022). According to this research, the predictive component foresees a user’s music preference based on their past music stream history, while the prescriptive component uses these predictions to create an optimally diverse recommendation list (Gurobi Optimization, 2022). This mixed approach to predictions is highly transferable to predicting course enrollment as it involves a lot of constraints, that can be enforced using Linear Programming, and the predictive analytics to make the predictions.

## Solution Space

This paper’s solution to the problem statements listed prior considers the technical limitations first. Our system’s primary goal is to minimize and predict waitlisting times, so the following tools will be used to accomplish this task:

* AWS for cloud platform and related services
* Gurobi for optimization problems
* Python and its related libraries for general coding, analysis, and automation

In terms of the Machine Learning Techniques, the ideal approach will be to conduct EDA (Exploratory Data Analysis) to find out the underlying patterns, and determine the scope and domain of the analytical methods to use. The most common EDA procedures will be:

* Univariate Analysis
* Correlation Study
* Data Bias and Representation Tests
* Diversity Metrics Tests

These tests will help in determining the most appropriate technique/algorithm to use for making predictions. The literature review study indicates that the analysis is very sensitive to the algorithms used, the data used to perform analysis, as well as the model tuning approaches undertaken to personalize the predictions for the use case. Thus, it makes sense to design various experiments to check the applicability and efficacy of each algorithm and technique in making the predictions. More experiments can be conducted in an iterative manner to further improve those models, or discard them based on the results obtained from previous experiments. Some of the techniques shortlisted for initial consideration are:

* Regression Models
* Tree Models (CART and Random Forest)
* Linear Programming and Optimization Models
* Time Series Analysis (ARIMA / S-ARIMA)
* Ensemble Models

In terms of cloud implementation, the expected size for the input is likely going to be under 1GB of information, but it could be larger dependent on the data scraped. Security is an issue for this project, as private student information is under the input folder. Those without proper access should not be allowed in, and this should be a programmable feature for those who are developers looking to improve upon this iteration. A scheduler should also be provided to our system that triggers whenever a new input file is introduced. AWS provides a resolution to both issues through its services. AWS’s Internet Gateway would allow for public subnets to access the internet and manage the total load for the site. Public subnets here are defined as information that the average user has access to, whereas private subnets focus on information that the system itself has access to.

The public subnet’s job here is to primarily check whether S3 bucket information has been updated by accessing it through it’s role permissions. This is an automatic process, so AWS Lambda will be called to start and call an EC2 instance in the public subnet. The public EC2 instance will use this as a time to scrape any data from the CEC website. That data is routed to a private PostgreSQL database within our database. Following the insertion of the scraped data, the EC2 subnet instance analyzes the data. It filters out information derived from both the scraped data and the input file, creating value constraints. These constraints will limit special cases and the track total number of pre-requisites. An evaluation will be processed through Gurobi for the expressed purpose of minimizing the waitlist. Gurobi is used in the public subnet as opposed to the public subnet because an ENI is required for EC2 instance to function properly. Once the data is collected, data from the prior dataset will be fed through an ARIMA and Linear Regression model to determine time series data and variable importance. The information is then fed back into the S3 storage unit, with all resultant analysis for that time period listed.

The diagram listed below provides a simplified list of the architecture this paper is hoping to develop:

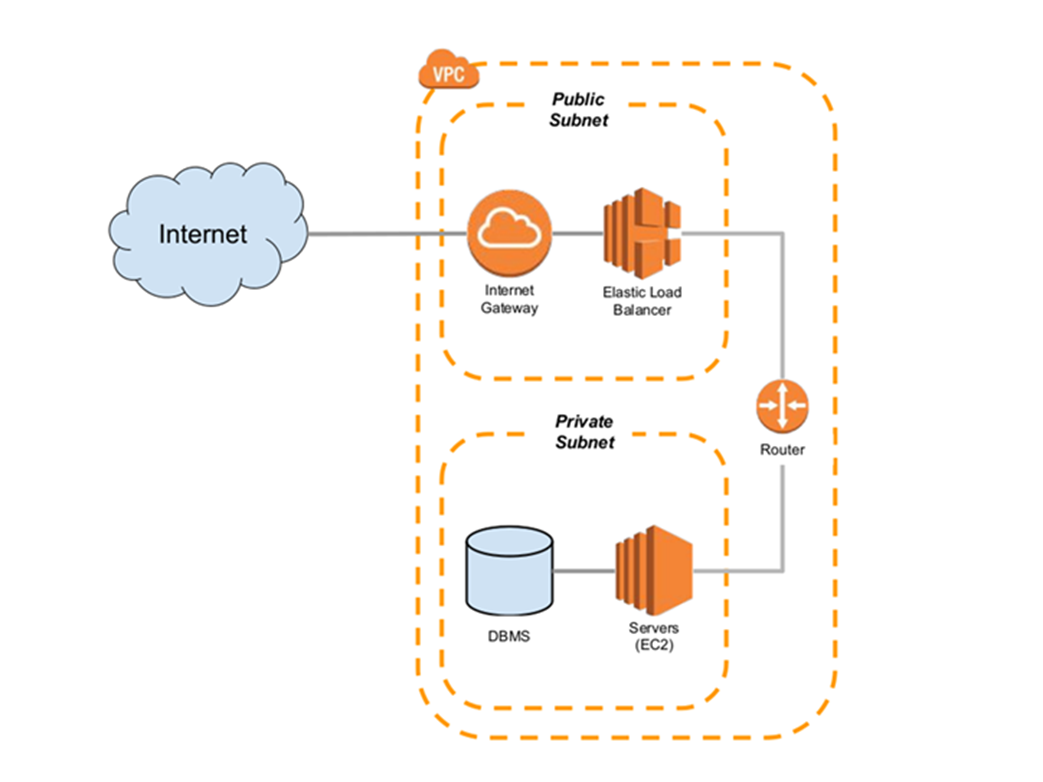


Figure Simplified Cloud Architecture for the Project

## Project Objectives

This project seeks to predict the future demand of courses in the College of Engineering and Computing (CEC). We are looking to analyze data provided by CEC Enrollment Management to find trends that achieve this goal for future courses and determine how many sections are necessary. Once we understand these trends, we will construct a dashboard for enrollment management to receive recommendations on which classes to offer. Enrollment Management will be able to upload files to keep data current, see recommendations of what courses have to offer, and how many sections should be available for each course. With this knowledge in the hands of Enrollment Management, they can better serve the student population to minimize waitlist numbers and provide students the classes they most desire. Addressing this issue will allow students to graduate on time and enable enrollment management to be better prepared to address the changing needs of their students.

## Primary User Stories

The following user stories is capture the core of our requirements gather process based on our primary objectives:

* As an educator, I want to be able to access the predictions of course enrollment demands so that I can make informed decisions about course scheduling and resource allocation.
* As the head of a course department, I would like to know the demand for courses in advance so that I can accommodate enough resources (classrooms, lab, professors, budget) to fulfill demand.
* As an enrollment management, I want to use Enrollment Prediction Utility to straightforwardly figure out the number of classes that the colleges should offer during upcoming semesters.
* As a student, I like to be able to register for my desire courses without being waitlisted so that I can finish my program on schedule.

## Product Vision

### Scenario #1

Students taking the CEC courses have been facing issues in registering for their desired courses. These waitlist lead to students not being able to register for a course, and thereby waiting for yet another semester in hopes of enrolling for the course. Whereas international students have to settle with other courses to maintain their Full-time enrollment status. Our work on Enrollment Prediction Utility can predict the demand of such courses well in advance, giving plenty of time for the CEC Department to accommodate the required resources (like classrooms, labs, faculty, budget, etc.) to fulfill the demand.

### Scenario #2

The Course demand for the CEC Courses fluctuates a lot, and it is becoming more difficult to make predictions. This results in a lot of courses that either have a very large waitlist, or too few registrations. Moreover, it is very difficult to make arrangements for new sessions, and allocate classrooms, budgets, and faculty near the start of the semester. Our work on Enrollment Prediction Utility will use the data available with the Enrollment Management at Office of the Provost to predict in future the demands for all the courses. The insights from these can help the CEC Department in figuring out the steps required to fulfill the said demand, well in advance. Moreover, it can also help professors in determining the appropriate modality for those courses, based on the number of student enrollments.

# Datasets

## Overview

**INSTRUCTIONS**

Provide a descriptive overview of your datasets.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Field Descriptions

**INSTRUCTIONS**

Described your dataset field. Make sure you study the example below and you will more than likely expand these fields:

1. URL (Type: string) – The web address or Universal Resource Locator for the webpage that contained the news article. This includes the protocol (http or https), host name, and subdomain. Some URLs also include parameters (text following ‘?’) or named anchors (text following a ‘#’). Each URL can only be present once in the database, even if the webpage is not static over time.
2. Title (Type: string) – The title of the news article as parsed by the Newspaper 3K module. This field may be null (~150 articles in our dataset do not have titles).
3. Authors (Type: string) –The authors of the news article as parsed by the Newspaper 3K module. This field may be null (~23,000 articles do not have authors) and articles with multiple authors have their names joined with a comma into a single string. This field may also pick up descriptions of the author, including their titles and background.
4. Publication Date (Type: datetime) – The article publication date and time as parsed by the Newspaper 3K module. The datetime is displayed in ISO 8601 format (YYYY-MM-DD Thh:mm:ss+offset). Publish dates without specified times are assumed to be published at midnight. Publication dates with time information, but without a time zone listing, are assumed to be in Eastern Standard Time. This field is not allowed to be null.
5. Text (Type: string) – The text of an article as parsed by Newspaper 3K. This field may be null (~8,000 articles do not have text) as some news stories are delivered as only video, audio, or a picture. The mean word count for text is 538.9 across all news sources.
6. Tags (Type: string) – Article tags as determined by Newspaper 3K. These appear to be important (rare or “topicy”) words taken from the article text, not meta tags contained in the article’s HTML. Multiple tags are concatenated with a comma into a single string.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Data Context

**INSTRUCTIONS**

Provide a description of the data context.

Data context is the set of circumstances that surround a collection of data. Capturing and interpreting context is a basic step in data analysis. Use of out-of-context data is a common source of errors in scientific research, business decisions, and professional advice.

In business analytics (BA), gathering context from external sources can provide useful information about events that have significance for the organization. Context for an unexplained surge in sales, for example, could be provided by pulling in data from news and social media as well as less obvious sources, such as weather over that period. Explored in context, it may be able to identify external causes for the increase, and that information might be used to guide future business decisions.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Data Conditioning

**INSTRUCTIONS**

Describe the data conditioning required for each data set.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Data Quality Assessment

**INSTRUCTIONS**

At a minimum you must assess your data sets with the following attributes:

* Completeness
* Uniqueness
* Accuracy
* Atomicity
* Conformity
* Overall Quality

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Other Data Sources

**INSTRUCTIONS**

If you are considered other data sources, however, you decided not to use these sources provide some reason why they were not utilized.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Storage Medium

**INSTRUCTIONS**

Discuss the storage medium selected for the project data set storage.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Storage Security

**INSTRUCTIONS**

Discuss the storage security required for the project data set storage.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Storage Costs

**INSTRUCTIONS**

Discuss storage costs associated with the storage medium used for the project data set storeage,

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

# Algorithms & Analysis / ML Model Exploration & Selection

## Solution Approach

**INSTRUCTIONS**

Provide a detailed discussion of the solution approach. Include discussions on any of the following:

1. Systems Architecture
2. Systems Security
3. Systems Data Flows
4. Algorithms & Analysis
5. Machine Learning (delete this subsection for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

### Systems Architecture

### Systems Security

### Systems Data Flows

### Algorithms & Analysis

## Machine Learning

**INSTRUCTIONS**

For Machine Learning projects discuss the model exploration and selection process. Delete this report subsection for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

### Model Exploration

### Model Selection

# Visualizations / ML Model Training, Evaluation, & Validation

## Overview

**INSTRUCTIONS**

Provide an overview of what was accomplished during Sprint 4. Focus visualizations for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

## Visualizations

## Machine Learning

**INSTRUCTIONS**

For Machine Learning projects, discuss your approach to the following with respect to the ML Model:

1. Training,
2. Evaluation, and
3. Validation of the ML Model.

Delete this report subsection for non-machine learning projects.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

### Model Training

### Model Evaluation

### Model Validation

# Findings

**INSTRUCTIONS**

Discuss the major findings of the project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

# Summary

**INSTRUCTIONS**

Summarize the overall project and results for the reader. What did you discover, prove, disprove, etc.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

# Future Work

**INSTRUCTIONS**

This is critical section of the report. Propose future follow-on work or next step(s) for the project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Appendix

Appendix A: Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Agile | Agile is a set of rules/philosophical ideas on how to develop and manage programming/coding projects. |
| Enrollment Management | Enrollment Management is a set of activities performed in acquiring new students, and managing student admissions, enrollment, and graduation processes in the higher level and professional degree issuing universities. |
| Linear Programming | Linear Programming is a technique for solving optimization problems, by drafting an optimization function, setting up variables, and subjecting it to constraints, all using linear equations. |
| Machine Learning | Machine Learning is a technique of enabling computers to learn patterns from examples/data, and then inferencing them to generate predictions, perform classifications, and make smart decisions. |
| Scrum | Scrum is a product development methodology that uses Agile principles to deliver the project in iterative steps, instead of traditional waterfall approach, where the project is undergone in a series of sequential steps. |

**INSTRUCTIONS**

Place all terms which require definitions in the Appendix A: Glossary.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Appendix B: GitHub Repository

Overview

**INSTRUCTIONS**

Provide a GitHub Link and the README.MD content. Do not just provide a link to the GitHub repository but provide a narrative paragraph which introduces the project. This section should mirror the look and feel of a well-documented professional GitHub site.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

GitHub Repository Link

https://github.com/GoswamiSagarD/Team-Prophecy

GitHub Repository Contents

Appendix C: Risks

Sprint 1 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Name | Description | Probability | Impact | Mitigation |
| Miscommunication | The team gets confused as to what is expected. | Medium | High | Encourage YouTrack to be checked and updated daily. |
| Task Management | The substantial number of tasks can be overwhelming. | Medium | Medium | Organize YouTrack to provide more guidance on tasks to be completed. |
| Scheduling Issues | It is a challenge to get everyone together at one time. | High | Medium | Encourage independent work and communicate work being completed |

Our team has faced a few risks while working toward completing Sprint 1 that negatively affect our progress. Miscommunication and Task Management are two similar issues we are working to mitigate. Miscommunications have caused confusion as to what tasks need to be completed. We communicate through various methods, and that could lead to confusion as we must trace what was said and agreed on. Task Management has been a challenge due to everyone being new to the scrum method and making sure the massive number of tasks are being addressed. These two issues along with scheduling issues can combine to allow team members to lose interest or disconnect from the work being done while working closely with everyone with the time constraints mitigates all the issues involved in this project.

Sprint 2 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 3 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 4 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 5 Risks

**INSTRUCTIONS**

Include the risk table associated with the Sprint. Below the risk table provide a narrative description of how the risks and mitigation plans were identified, what the team got correct, what the team could have done differently, how accurate was the team in identifying the risks, did the team encounter any unanticipated risks, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Appendix D: Agile Development

Scrum Methodology

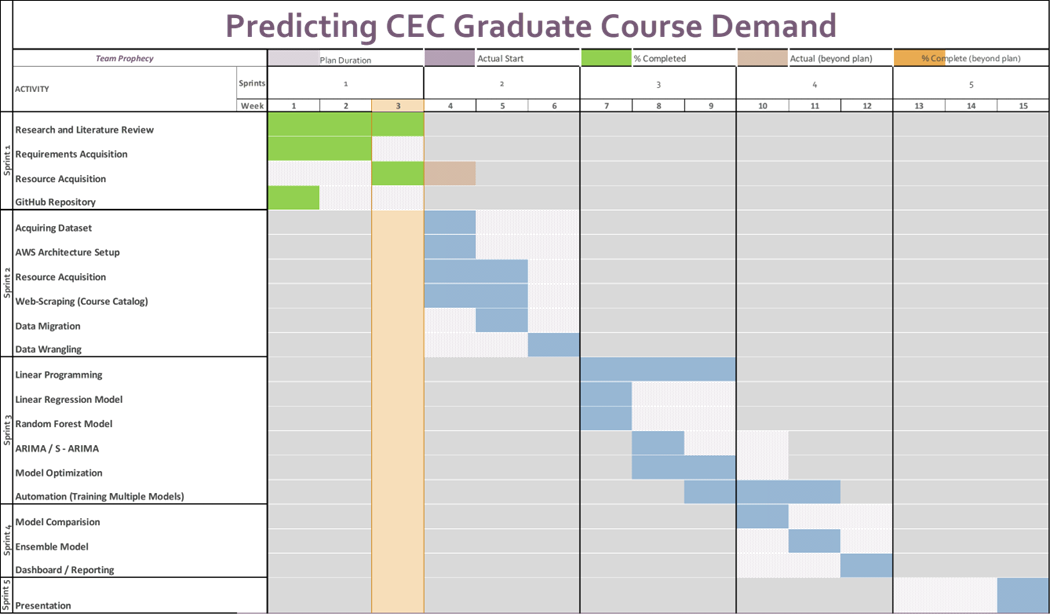


Figure Sprint 1: Scrum Planning Board

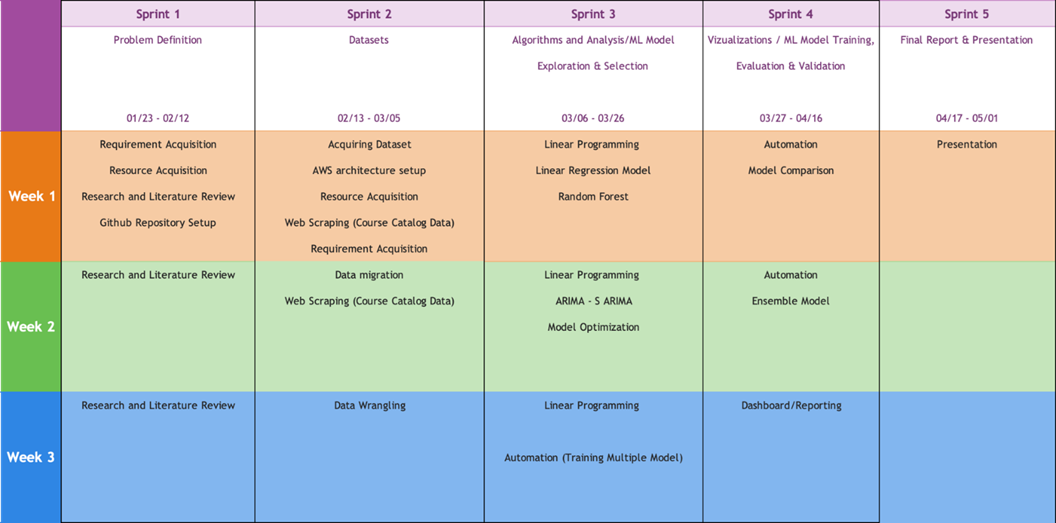


Figure Sprint Planning Sheet

Sprint 1 Analysis

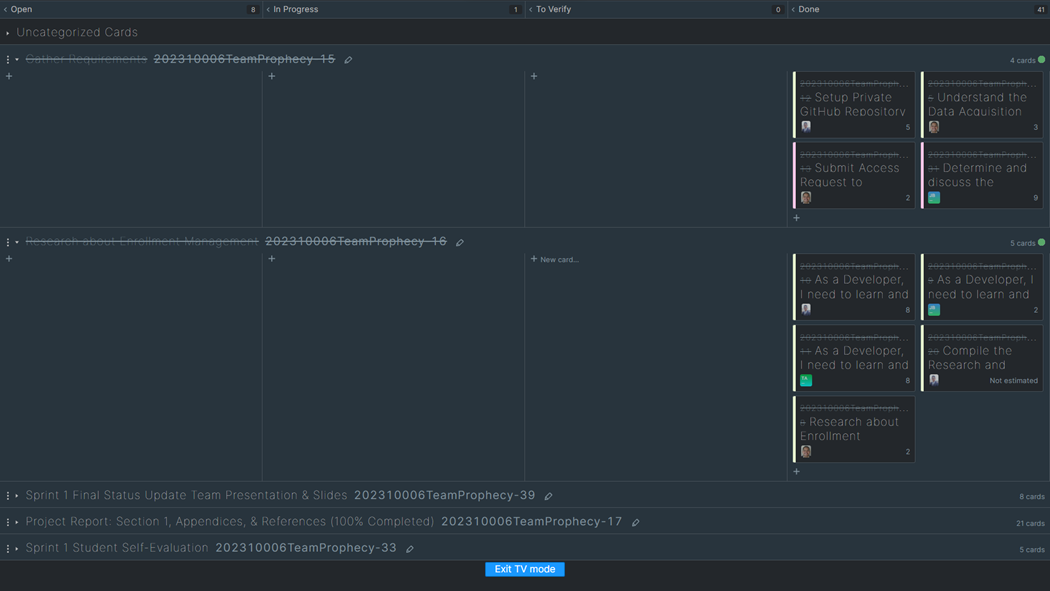


Figure Sprint 1: Agile Board

Sprint 2 Analyis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 3 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 4 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Sprint 5 Analysis

**INSTRUCTIONS**

Provide a narrative of the team’s efforts during this Sprint. Be sure to include – but not be limited to – how the team identified the User Stories, how well the team performed with the various tasks, how easy/difficult it was for the team to manage their activities during the Sprint, what did the team do correct, what could/should the team have done differently, etc. Think of this writeup as a “lessons learned” that you would like to pass along to any project team thinking of doing a similar project.

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

Reference

Works Cited

Data USA. (n.d.). *George Mason University.* Data USA. Retrieved from https://datausa.io/profile/university/george-mason-university/#:~:text=George%20Mason%20University%20received%2021%2C198%20undergraduate%20applications%20in,accepted%20for%20enrollment%2C%20representing%20a%2089.2%25%20acceptance%20rate.

Gurobi Optimization. (2022). A Music Recommendation System with Mathematical Optimization. Retrieved from https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=GJBs\_flRovLc

Gurobi Optimization. (2022). Music Recommendation System: Jupyter Notebook Model for a Music Recommendation System. Gurobi Optimization. Retrieved from https://www.gurobi.com/jupyter\_models/diverse-music-recommendation-system/

Hassan Bousnguar, L. N. (2022, August). Forecasting approaches in a higher education setting. *Education and Information Technologies*, 27. Retrieved from https://link.springer.com/article/10.1007/s10639-021-10684-z

Shao, L., Ieong, M., Levine, R. A., Stronach, J., & Fan, J. (2022). Machine Learning Methods for Course Enrollment Prediction. *Strategic Enrollment Management Quarterly, 10*(2), 11-29. Retrieved from https://www.proquest.com/docview/2697182295/fulltextPDF/A3E7B94D67954EA8PQ/1?accountid=14541

Soltys, M., Dang, H., Reilly, G. R., & Soltys, K. (2021). Enrollment Predictions with Machine Learning. *Strategic Enrollment Management Quarterly, 9*(2), 11-18. Retrieved from https://www.proquest.com/docview/2606939441/fulltextPDF/6C409A77790B4A22PQ/4?accountid=14541

**INSTRUCTIONS**

The References section of this document makes use of the Microsoft Word References feature to insert research citations by recording them directly into the document. All citations are to follow the IEEE citation format. Use the Bibliography drop down to have Microsoft Word dynamically create your Works Cited section

here in IEEE citation format.

To learn more about the IEEE Citation guidelines click on the document links below.

1. [IEEE-Reference-Guide.pdf](https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf)
2. [IEEE Citation Guidelines2.doc (ieee-dataport.org)](https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf)

**DELETE THIS TEXT BOX AFTER YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS.**

**This page intentionally left blank**