

Jikan Control

Software Engineering Project (Capstone) - Final Report

Developers:

- Axviel A. Benítez Dorta - INSO 4151
- Fernando Guzmán Fernández - INSO 4151
- Angel Rivera Camacho - CIIC 4151

Mentor:

- Professor Kejie Lu

Table of Contents

Table of Contents	2
Executive Summary	3
Introduction	4
Problem Statement	4
Objectives	5
Summary of Solution	7
Market Analysis	8
Related Works	9
Machine Learning Model Analysis	10
Technical Description	17
Constraints	25
Project Plan	26
How It Was Implemented	27
Performance	29
Projected Costs	30
Testing	30
Pros and Cons	31
Lessons Learned	31
Future Work	31
Applied Standards	32
Links for Jikan Control	35
References	37

Executive Summary

Jikan Control is an application that can be treated as an intelligent calendar for students. A calendar is a tool used for organizing days and hours into events and activities. Usually, a student's performance in a course is measured with several exams, which can be considered events. This application is able to predict the score that a student will receive in any given exam and how much time they need to dedicate to studying to get a specified grade. This is done with the help of a machine learning model created using a dataset which contains correlations between study time, absences, and exam performance. Eventually the application will use the information obtained from the users to enlarge the dataset and improve the Machine Learning model's accuracy. The application analyzes the amount of time that a student has dedicated (or will dedicate) to studying for an exam and predicts the result that the student will receive.

In addition to predicting the student's performance, this application can identify how much time a student has to spend studying in order to get a specific exam result. This is important when determining the optimal amount of time that should be dedicated to multiple exams that are taken in a similar time period.

Jikan Control's main user interface feature is the calendar. In this part, the user can view and add events. An event can be an exam, study time period, reminder, and any other regular activity such as the period in which an academic course is being given. These events can also have one or multiple notes attached to them.

Finally, this is a web application, so it can be accessed from any desktop, laptop, and mobile device with a supported web browser. It is hosted on the Heroku platform.

Introduction

College students lead an interesting life. They go to a campus where they take multiple courses, sit down and study for said courses, go out and party with their friends, and travel outside of the local university area. This simplified view of college life shows the many ways in which a student's time can be spent. In order to manage these activities the use of an agenda is encouraged. It allows you to increase your productivity by organizing your time.

The bane of a student's existence most of the time is the exams that they take in these courses. It can be difficult to determine how much time one needs to allocate for studying for an upcoming exam. This study time may vary depending on how much of the material the student has available, the period of the semester in which the exam is being given, how many other exams are to be taken in the same short time period, and most importantly how much free time the student has available.

Problem Statement

A student can use a calendar for creating reminders for upcoming events. It can also be used to create relevant notes for any event. However there is something that a regular calendar cannot do: predict the scores of an exam. This major problem is what this software project aims to solve. Can a student's performance in an exam be predicted using information such as study time and number of absences? Also, is it possible to calculate the amount of study time needed to receive a particular score in an exam? Given enough data and the correct machine learning model, it certainly seems possible to do both of these things.

Now, why does this matter? Can't a student just study as much as possible in order to receive the highest score possible? Not really. Usually students take multiple exams in the same week or even on the same day. In addition to that, they also have social lives and family to keep up with. It is not realistic to expect a student to dedicate every part of their free time to studying. That is where this application comes in.

A student can determine the optimal amount of study time that will result in the best possible results given their unique circumstances. How this will be determined is with the use of machine learning models. By obtaining an existing dataset that contains correlations between study time, absences, and exam scores, a machine learning model can be created in order to predict exam scores based on a given number of study hours and absences.

Of course, this will be helpful to students. Students are the type of client that we aim to provide this service to. Full time and non full time students alike could benefit greatly from this application.

Objectives

The main objectives for this project were the following:

1. Create a fully responsive User Interface.
2. Create a backend system that connects to the frontend system.
3. Create a database system.
4. Obtain an existing dataset that contains a correlation between study time (and other inputs) and the score received in an exam.

5. Research different machine learning supervised models and determine which model will best work with the found dataset.
6. Create a custom calendar.
7. Develop a suite of automated UI tests.
8. Develop a notification system that reminds the user of upcoming events.
9. Deploy the application in an online platform.

These objectives can be considered completed when:

1. The UI allows the user to fully use the application on any device. A complete use of the application includes creating accounts, events, exams, notes, and predicting exam scores. Such UI was constructed with HTML, Bootstrap for CSS, and the Jinja Template.
2. The backend system takes all the user input data, stores them into the database, and returns any type of machine learning results and queries. The backend system was constructed with Python and the web framework Django.
3. The database can store all user data and run complex logic. The database that was chosen for the project is PostgreSQL and the complex was stored in Postgres functions.
4. The obtained dataset is large enough to construct good machine learning models. The dataset used was the 'Student Alcohol Consumption' dataset created by UCI Machine Learning.
5. Develop an in depth analysis that justifies why the chosen machine learning model is the best option for the dataset. After completing a machine learning model analysis, the linear regression model was determined to be the best choice.

6. The calendar displays all of the events that a user creates. The created calendar contains two views: Month and Week.
7. The tests simulate most of what a user will do in the application. The tests were created using Selenium Webdriver.
8. The application delivers a desktop push notification when the event starts.
9. The deployed version of the application can be accessed by any person in the world. The application is currently hosted on the Heroku platform.

Summary of Solution

The solution that is being proposed is a simple web application that holds a student's calendar and predicts the score that a student will receive on an exam based on their study time and number of absences. Also, the student can receive a predicted value for how much time they should study in order to receive a desired score. Also, this application can serve as a replacement to any student's current calendar, whether it's the Google Calendar, Outlook Calendar, etc. Especially since none of these existing calendars can provide the predictions that this application provides.

The benefits for the users are straight forward: easier and better time management, and better academic performance. In addition, the users also have an economic benefit. When this is taken into consideration, then this application becomes very attractive. Due to this being a relatively small application, the price or subscription fee of the product will be student friendly, or even free with some advertisements. Finally, due to the massive amounts of students that would benefit from an application like this, the developers could earn a decent amount of money.

The main challenge in successfully commercializing this application is just raising awareness of its existence. In the University of Puerto Rico Mayaguez system there are over 10,000 students that take classes each semester. Several thousands of students are also in other UPR systems and the same can be said for private universities. And that's just in Puerto Rico. A low price of \$5 or \$10 is enough to incentivise enough students to adopt the application for a few months, a semester or their entire college life.

Market Analysis

The main market focus of this application is in the academic field and the students. Currently these students rely on some calendar services like Google Calendar or Outlook Calendar to keep track of their exams, homeworks or other deadlines from their courses. What makes this application attractive to students is that it was specifically designed for them and their needs when it comes to academic performance.

The biggest risk that arises from the application is in the event of unauthorized access to a user's account. This would mean that their privacy and security is at risk since now bad actors have access to the user's schedule. However this is the same risk presented in the Google Calendar, Outlook Calendar, or any other similar product.

To get this application operational the only thing that's really needed is a server to host the application and database. This is something that is really flexible since there are many web hosting services and most of them are flexible on what type of applications they can host.

There doesn't seem to be much work done in the field of predicting exam scores, so this part of the application could be treated as intellectual property. With further work and improvement on the predictions, one way to protect this IP is by creating a patent for it.

Related Works

Research was done to determine how to create a machine learning model that would predict exam scores. While researching, information was found that suggested that exam scores could be predicted using linear regression. In addition, a dataset containing the data needed for the predictions was found in the website Kaggle.

In the references of this report there are many resources that were used to learn all about machine learning. There is an article found written by Terence S and published in towardsdatascience.com that talks about what is a prediction and the many types of models that exist. Using this resource it was clear that regression models were the way to go with this dataset. Then there is a research paper by Hazlina Darman that mentions the success that they had in predicting final scores using the linear regression model, and with further investigation it was found that the sklearn (a python toolset for machine learning) made this type of linear regression model much easier to implement. There aren't other papers investigating the prediction of exam score with the dataset that was found for this project and the data that the other investigations had was not readily available, therefore in this report the *Machine Learning Model Analysis* section goes into depth about the findings that were made during this semester with this dataset. In addition to predicting exam scores the prediction of study time was also investigated, this is something that the other papers did not touch on.

Finally, the project's main feature is a calendar and it's the thing that users will mostly interact with. Inspiration was taken from the Google and Outlook Calendar (who allow for events and notes to be created) when designing the Jikan Calendar, with some added functionality that is required when creating a calendar designed for students (like the machine learning component).

Machine Learning Model Analysis

The Dataset

The "Student Alcohol Consumption" dataset by UCI Machine Learning was used to create the machine learning model. This dataset was created for the purpose of determining if alcohol consumption along with other factors had any effect on academic performance. The dataset is made up of two smaller datasets. One measures the performance in a math course, while the other measures performance in a language course. The students are between 15 to 20 years old. Exam score values are found between 0 and 20.

The Model

Based on conventional knowledge, it was assumed that the longer a student studies for an exam, the better they will perform. In addition, the better they have done in previous exams then the better they will do in the next exam due to their clear understanding of the course material. Finally, the less that a student is absent from class, the better they will perform.

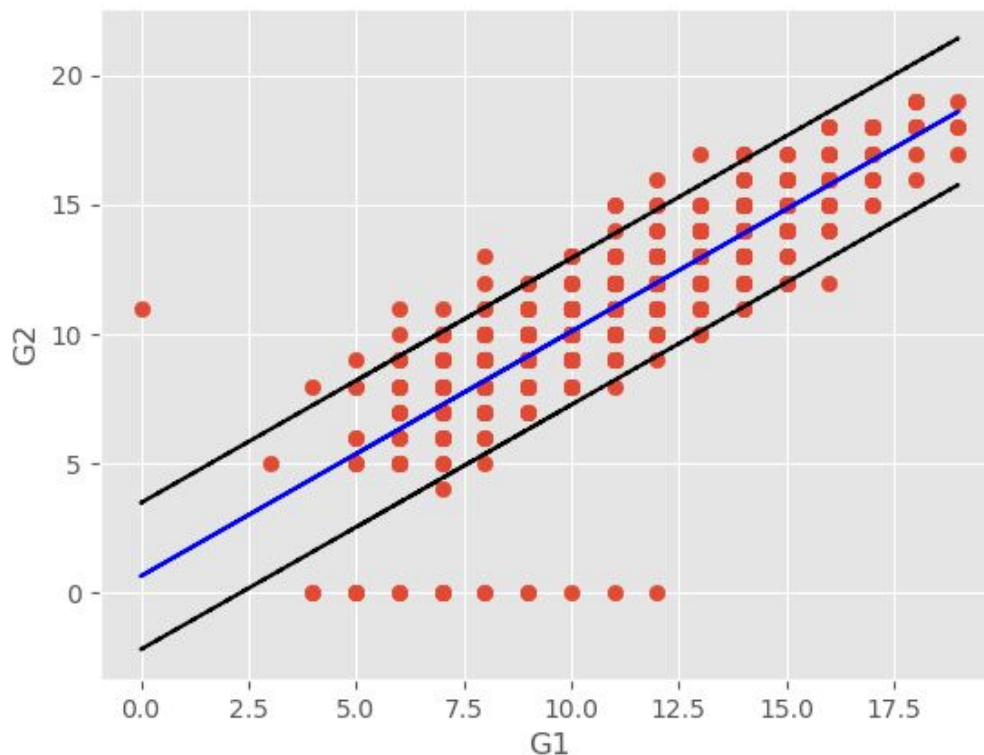
Based on the relationship observed using these assumptions, it was decided that a "linear regression" model would be used to construct the machine learning model of the application.

However, these are only assumptions. The data must be analysed and these assumptions must be proven in order to determine if linear regression is right for this dataset.

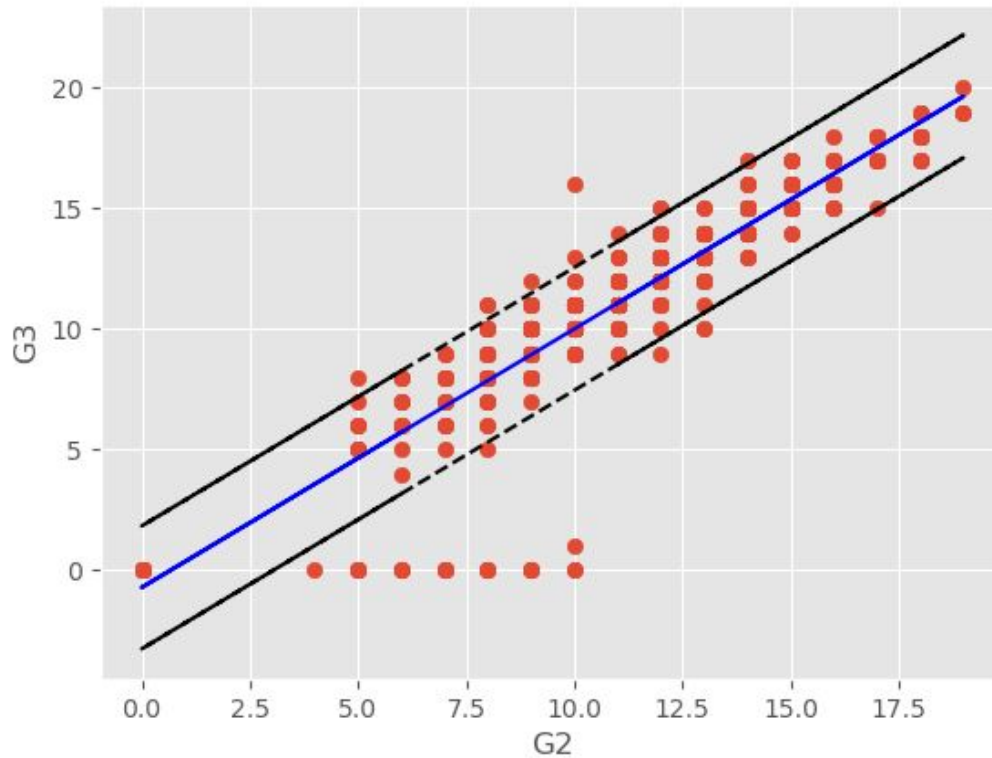
Dataset Analysis

The assumption that there is a linear relationship between the previous exam performance and the next was proven correct. This can be seen by plotting the relationship between the first exam (G1, x-axis) and the second exam (G2, y-axis) or by plotting the second exam (G2) and the third exam (G3). Most of the students that received a high score on the first exam, also received a high score on the second exam and vice versa.

The equation for the fitted line is: $y = 0.9456x + 0.6429 \pm \sigma$ where **variance** (σ) = 2.83

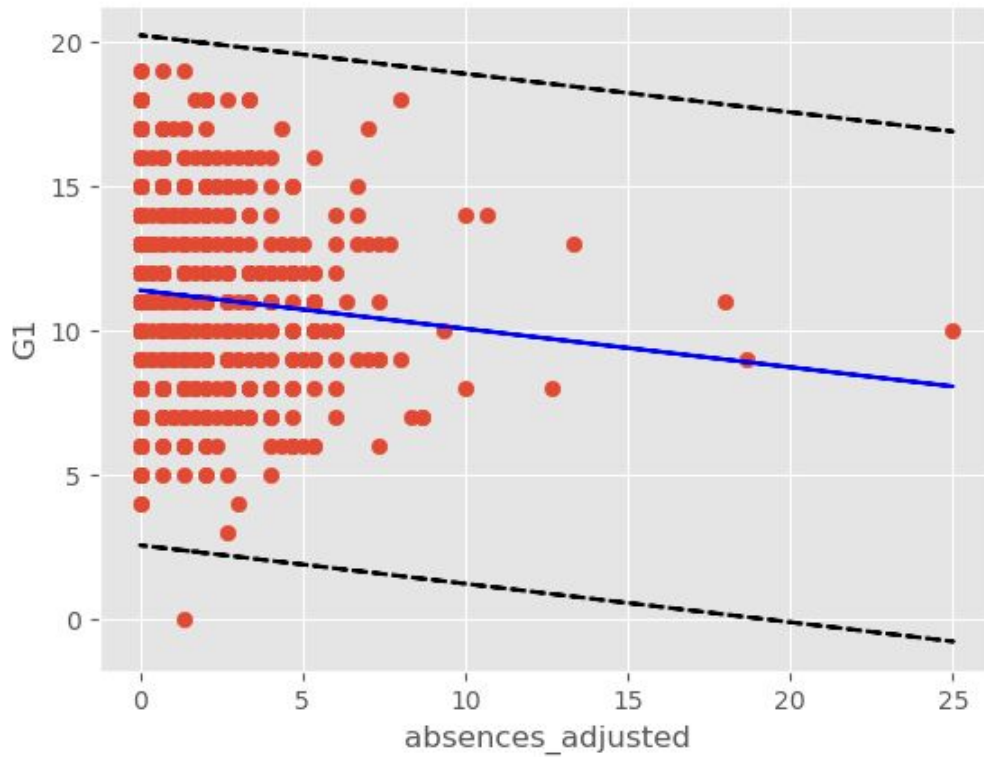


The equation for the fitted line is: $y = 1.0715x - 0.7079 \pm \sigma$ where **variance** (σ) = 2.55



The relationship between academic performance and the number of absences was also found to be linear. However it is an inverse relationship. The fewer absences a student had, the better they performed in the exams.

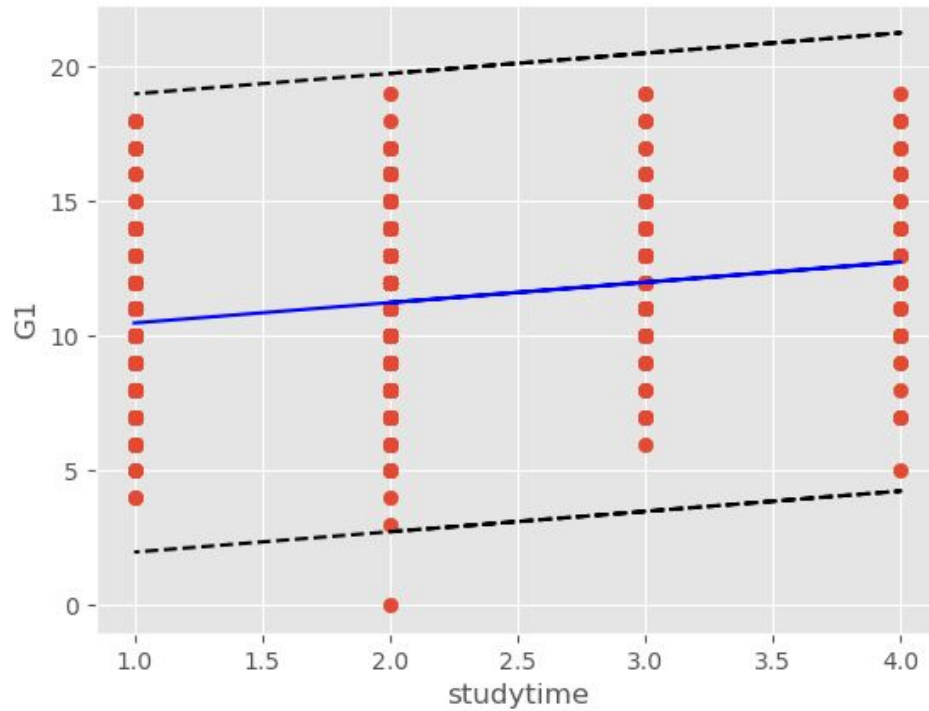
The equation for the fitted line is: $y = -0.1332x + 11.4105 \pm \sigma$ where **variance** (σ) = 8.82



Finally, the relationship between study time and exam score was the most interesting.

The number of study hours did not have the impact that was expected. More study hours did result in a higher exam score, thus showing that there is a linear relationship. However the rate at which the exam score increased when study hours increased was very small.

The equation for the fitted line is: $y = 0.7556x + 9.7248 \pm \sigma$ where **variance** (σ) = 8.50



One of the explanations for this finding is the difficulty of the course material. It was previously mentioned that the students in the dataset were 15 to 20 years old. The difficulty of the material that students of this age receive is not as high as the difficulty that students of ages 21 or high receive. Therefore, one could assume that a student in this dataset does not need to study much in order to receive a grade that is 50% or higher. This can actually be seen when trying to predict the exam score for the first exam (G1) of a given course in the application.

Event*	Course*	Exam Number*
<input type="text" value="Exam Event"/>	<input type="text" value="The Course"/>	<input type="text" value="1"/>
Predicted		
Study Hours*	Absences*	Exam Score*
<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="53"/>
<input type="button" value="Predict Study Hours"/>		<input type="button" value="Predict Exam Score"/>

However for the first exam, an increase in study hours does have a significant impact on the predicted exam score. It is when predicting the score for the following exams that an increase in study time has less of an impact compared to the number of absences and the previous exam score.

Event*	Course*	Exam Number*
<input type="text" value="Exam Event"/>	<input type="text" value="The Course"/>	<input type="text" value="1"/>
Predicted		
Study Hours*	Absences*	Exam Score*
<input type="text" value="12"/>	<input type="text" value="0"/>	<input type="text" value="87"/>
<input type="button" value="Predict Study Hours"/>		<input type="button" value="Predict Exam Score"/>

Event*	Course*	Exam Number*
<input type="text" value="Exam Event 2"/>	<input type="text" value="The Course"/>	<input type="text" value="2"/>
Predicted		
Study Hours*	Absences*	Exam Score*
<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="85"/>
<input type="button" value="Predict Study Hours"/>		<input type="button" value="Predict Exam Score"/>

Event*	Course*	Exam Number*
<input type="text" value="Exam Event 2"/>	<input type="text" value="The Course"/>	<input type="text" value="2"/>
Predicted		
Study Hours*	Absences*	Exam Score*
<input type="text" value="7"/>	<input type="text" value="0"/>	<input type="text" value="86"/>
<input type="button" value="Predict Study Hours"/>	<input type="button" value="Predict Exam Score"/>	

Analysis Conclusion

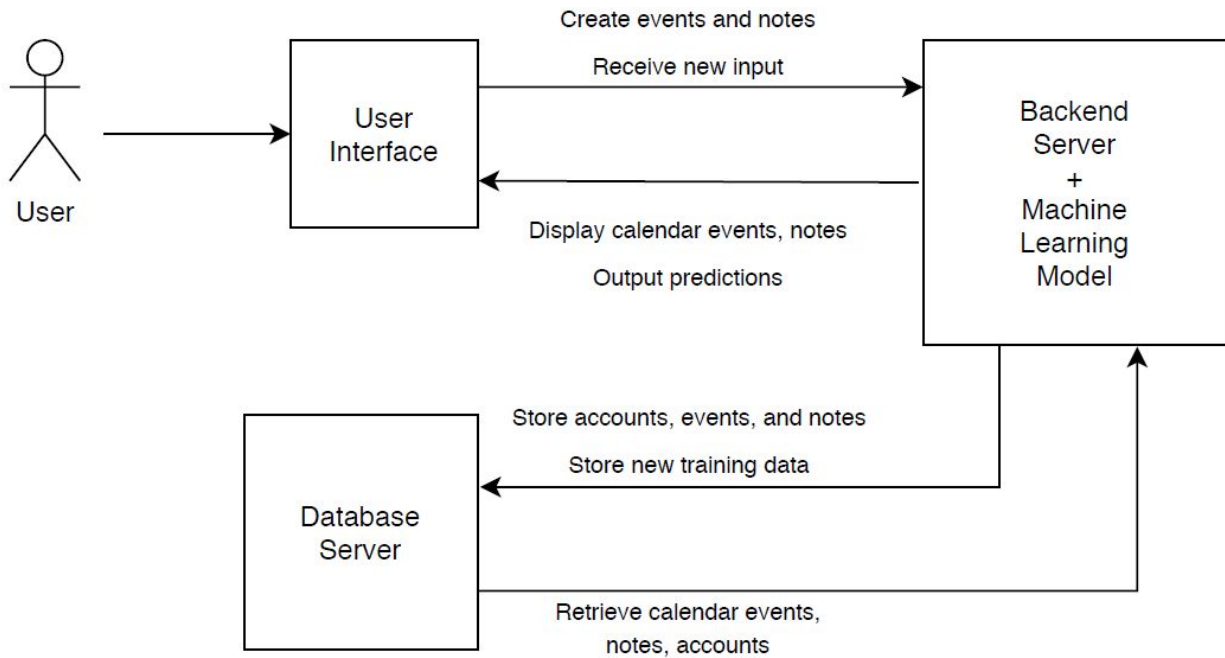
The model that best fits the dataset is a linear regression model. It is very effective at predicting academic performance when looking at the previous exam scores and the number of absences. While it may not be the best model for predicting exam scores based only on study time, it does a good job when all of the factors are taken into consideration. Variance is a measure of how far the set of numbers are spread out from the mean value. The lower the value, the closer we can expect values to be at the trend line. This variation is important to know since it indicates what values can be expected that aren't necessarily in the fitted line and we want a lower variance since that means that values vary little from the trend line. Looking at the variance values in each of the graphs we find that variance is very low in the graphs that compare grades (first two graphs), the next lowest variance is in the graph of G1 vs study time, then the highest variance is on the graph of absences and G1. The linear regression model focuses on the relationship between a dependent variable and a set of independent variables, the dependent variable being the outcome that we want to predict. To get a model that can accurately predict the grades and study time of exams we use all of the attributes graphed above to fit them into the linear regression model using the "hold-out" method. We split the data into a training set

and test set, the training set is used to train the model and the test set is used to test the accuracy of the model, with this method we get results which we can say we are satisfied with. To calculate this accuracy we use sklearn's *score* method which takes the split sets and calculates the r2 score of the model, and this also calculates the variance since it is used in the r2 scoring. In our testing this scoring method scored our G3 prediction model at 96%.

Technical Description

Jikan Control has three main components: frontend or client side code, backend or server side code, and the database. The frontend is constructed with Javascript, HTML, Bootstrap for the CSS, and the Jinja Template. The backend is constructed with Python Django. Finally, PostgreSQL is the database that is being used. The machine learning model that is being used is scikit-learn's implementation of a linear regression model.

System Architecture



Modules

The main modules (or Django apps) of this application are:

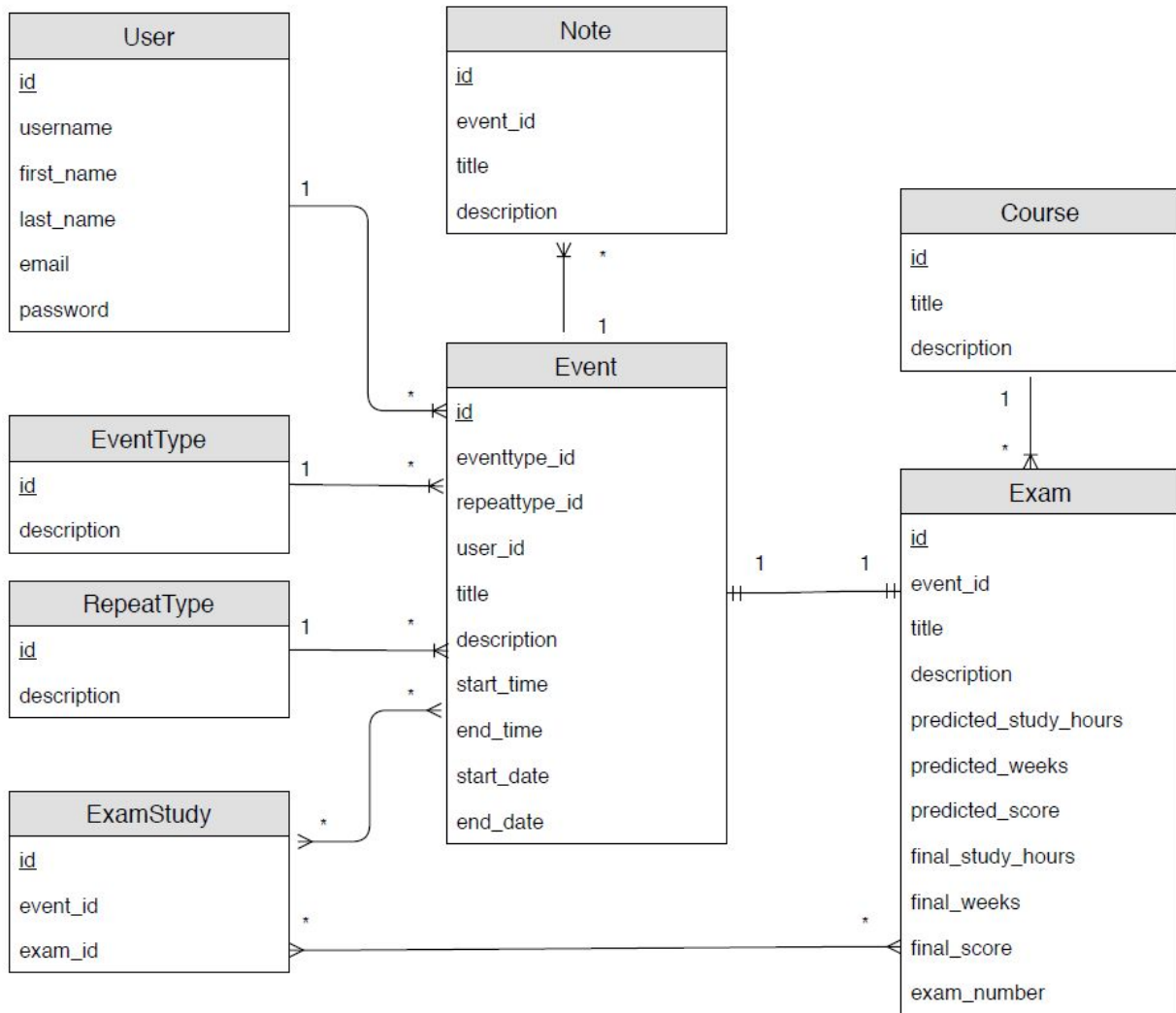
1. Accounts - Contains everything related to user accounts: sign up, login, logout, change password.
2. Calendar - Contains the events of a user and displays them in a calendar view.
3. Events - Contains the time in which the activity will take place. It also contains the type of activity (exam, class, reminder, etc).
4. Exams - Contains both the predicted and actual exam scores and study time periods.
5. Notes - Contains information that the user enters about an event.

In addition to the main modules, there are the supporting modules:

1. Courses - The course to which an exam points to. Used to determine what period the exam belongs to.
2. ExamStudy - Creates a relationship between an exam and the study events that were generated based on the machine learning model predictions.
3. EventType - Used to determine the type of event. Events can be: tasks, exams, studytime, and reminders.
4. RepeatType - Used to determine how frequently an event repeats. An event can repeat: daily, monthly, and yearly.
5. ML - Contains the function for calling upon the machine learning model for making predictions and adding to the model.

The “RepeatType” entity is needed in order to display events that repeat in the calendar without having to create individual events. The “Course” entity became a necessity for determining the period in which an exam was taking place. Finally, the “ExamStudy” entity solved the issue of deleting or updating generated study events when an exam predicted score or study hours were updated.

ER Diagram

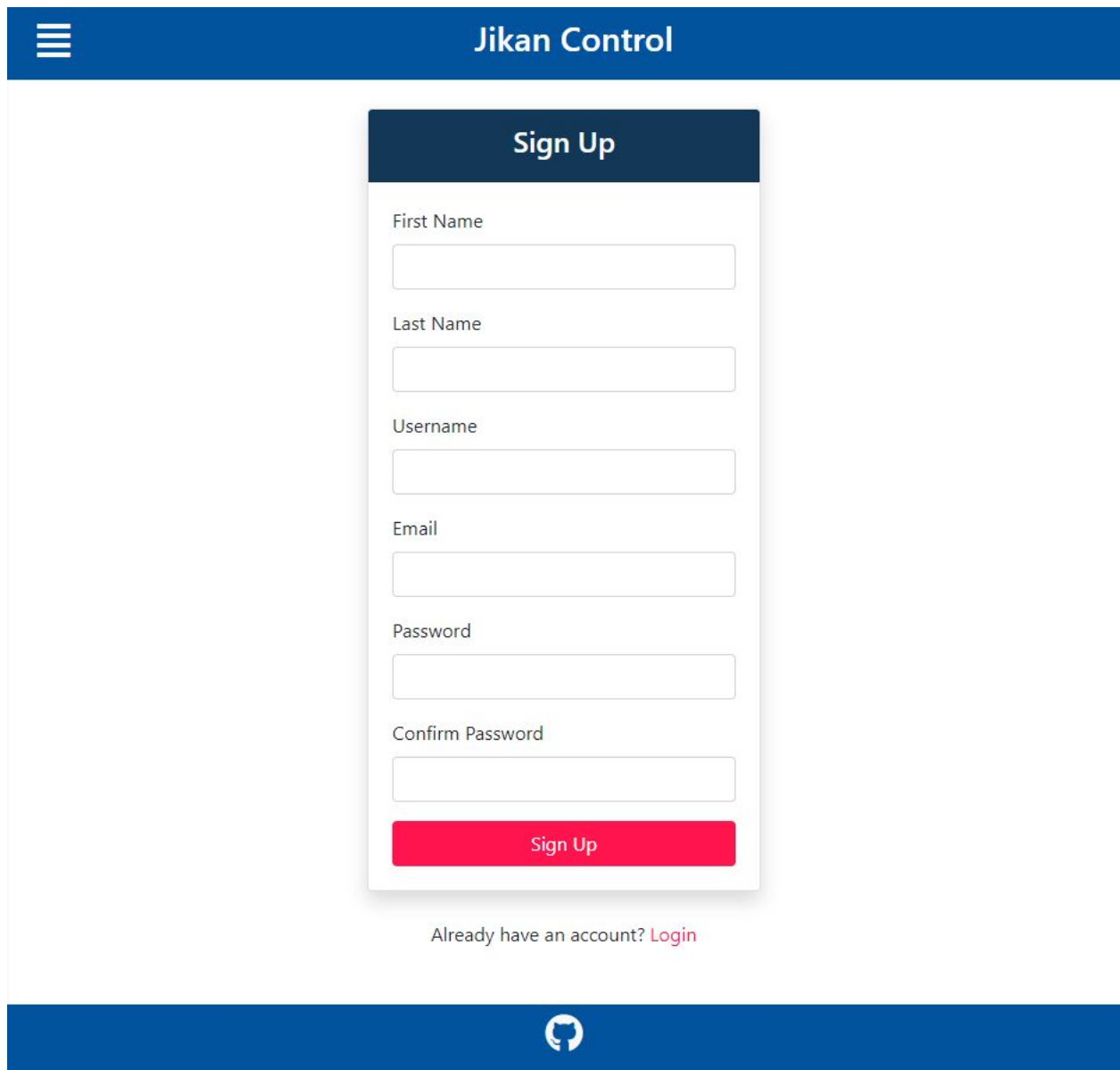


UI

The user interface of the system is made up of four main types of pages: authentication, calendar, list, and detail. The authentication group contains the Login, Password Reset, Password Recovery, and Sign Up pages. The calendar group only contains the Jikan Calendar with Month

and Week views. The list group contains pages that list all of the entities of the system, such as Events, Exams, Notes, etc. Finally, the detail group contains the pages that allow the user to edit the information of the created entities.

Authentication



The image shows a web application interface for "Jikan Control". At the top, there is a dark blue header with a white hamburger menu icon on the left and the text "Jikan Control" in white. Below the header is a white main content area. In the center of this area is a "Sign Up" form. The form has a dark blue header with the text "Sign Up" in white. Below the header are six input fields, each with a label above it: "First Name", "Last Name", "Username", "Email", "Password", and "Confirm Password". At the bottom of the form is a red button with the text "Sign Up" in white. Below the form, there is a link that says "Already have an account? [Login](#)". At the bottom of the page is a dark blue footer with a white GitHub logo in the center.

Jikan Control

Sign Up

First Name

Last Name

Username


Email

Password

Confirm Password

[Sign Up](#)

Already have an account? [Login](#)





Login

Username

Password

Login

Don't have an account? [Sign Up](#)

Forgot Password? [Reset Password](#)



Calendar

Jikan Control

Week Today ◀ ▶ September 2019

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Jikan Control


Home Events

Events

Search Events New Event

Title	Type	Start Time	End Time	Repeat	Date	Actions
Task N - Debug Calendar	Task	9 a.m.	10 a.m.	Does not repeat	March 27, 2020	i ✓ 🗑
Task C - Daily	Task	10 a.m.	noon	Daily	March 22, 2020	i 🗑
Task R - Completed	Task	5 p.m.	6 p.m.	Weekly	March 19, 2020	i 🗑
Event Exam E - Halp	Exam	8 a.m.	9 a.m.	Does not repeat	March 13, 2020	i ✓ 🗑
Exam F Event - Err Test	Exam	2 p.m.	4 p.m.	Does not repeat	March 12, 2020	i ✓ 🗑
Exam E Event - ML Test	Exam	4 p.m.	5 p.m.	Does not repeat	March 11, 2020	i ✓ 🗑
Exam D - Test ML	Exam	3 p.m.	4 p.m.	Does not repeat	March 10, 2020	i ✓ 🗑
Exam C Event - Limit Logic	Exam	2 p.m.	4 p.m.	Does not repeat	March 9, 2020	i ✓ 🗑
Task A - Monthly	Task	8 a.m.	10 a.m.	Monthly	Jan. 19, 2020	i ✓ 🗑
Task M - Monthly Past Year	Task	7 p.m.	8 p.m.	Monthly	Oct. 18, 2019	i ✓ 🗑

« 1 2 3 4 5 6 7 »



Detail

The screenshot displays the 'Jikan Control' application interface. At the top, a blue header contains a hamburger menu icon and the text 'Jikan Control'. Below this is a green breadcrumb trail: 'Home > Events > Task N - Debug Calendar'. The main content area has a dark blue header with the word 'Event'. Below that, the title 'Editing Event: Task N - Debug Calendar' is shown, followed by a blue 'Detail' button. The form contains several fields: 'Title*' with the value 'Task N - Debug Calendar', 'Description*' with the value 'Desc: Task N - Debug Calendar', 'Event Type*' set to 'Task', and 'Repeat Type*' set to 'Does not repeat'. There are also date and time pickers: 'Start Date*' (03/27/2020), 'Start Time*' (09:00), and 'End Time*' (10:00). At the bottom, there are three large, colored buttons: a green 'Save' button, a blue 'Complete' button, and a red 'Remove' button.

The list and detail pages for the other entities follow the same layout scheme.

Constraints

Time was the biggest constraint in this project due to it being developed in less than 4 months. Because of this the number of milestones and deliverables was not as large as it could

have been if there was more time. Another constraint is the dataset used in the machine learning models. This data belongs to students between the age of 15 and 20, while the application primarily targets college students that tend to be 18 years or older. This had an impact on the accuracy of the predictions of the first exam of a course. In addition the dataset set contains information that is very personal, such as relationship status and drinking habits. A significant portion of the users of this application wouldn't feel comfortable sharing that information. Due to that reason the information that this application stores from the users is only relevant to the current machine learning models. If every user consented to sharing that information, then it is possible that machine learning models with higher accuracy could be created.

Project Plan

Milestones and Deliverables

The key milestones that were delivered for this project:

1. Fully designed User Interface
2. Complete backend system
3. Complete database system
4. Machine Learning Model Analysis
5. Complete Machine Learning Model
6. Online Application Deployed Version
7. Application Video Demonstrations
8. Application Powerpoint Presentations
9. Application Manuals

The complete Gantt Chart for this report can be found here:

https://github.com/axviel/JIKAN_CONTROL/wiki/files/Final_Report_Gantt_Chart.pdf

Collaboration tools

The collaboration tools that were used in this project:

1. Github - remote repository to store all code created for this project
2. Google Docs - an online word processor that lets you work with other people.
3. Google Slides - an online presentation program that lets you collaborate with other people.

Presentation & Video Demonstration

The Powerpoint presentation of the application can be found here:

<https://youtu.be/P3BgCYOSk6g>

A video demo of some of the application can be found here: <https://youtu.be/TfRZkhNAq7w>

How It Was Implemented

UI

The User Interface was made with HTML and SASS (which was then compiled to CSS). The Jinja template was used to ‘connect’ the frontend and backend. It was also used to prevent code duplication by separating parts of the UI (i.e. the navbar) into files that are imported in every page. Most of the icons used on the application are provided by “FontAwesome.” The base code for the side menu was provided by the website “bootstrapious” and then customized to fit

the application. Finally, the base code for the calendar was provided by a Youtube user “Neat Developer.” This code was then further modified and enhanced to fit the application.

Server Side Code

Python Django was used to create most of the server side code. The site routes were created in the `urls.py` files. The methods that return data, render pages, and redirects to other pages were created in the `views.py` files. Finally, the forms used to create each of the main entities were defined in the `forms.py` files. Each Django app has their own set of the python files.

The Machine Learning Model was created with the `sklearn` library. Each of the models for predicting exam scores are created and stored in the same server.

The Password Reset feature of the application was created using Django’s built in password reset routes and views. An email account was created and linked to the application in the `settings.py` file. When a user requests a password reset, they receive an email from the linked account. The email contains a link to the application with a unique id and token. This allows the user to reset their password.

Push Notifications

The implementation of Push notifications has two parts, the service worker and the part that calculates the timer. The service worker allows us to send notifications from the django app to the user’s browser. There is a script that runs on the website that asks the user if he/she wants to allow notifications from us and when the user allows, a service worker is registered in the browser and we are able to send notifications. The other part is running on the server, this function is called every time the user logs in and every time there is a change to an event. This

function uses a combination of python threads and subprocesses. This function is called in a different thread so as to not make the rest of the application wait. The thread then fetches all of the user's events and calculates the time until the event happens and stores them in a priority queue so that they are ordered from most recent. Once this list is ready, the code makes a subprocess that its only job is to wait the amount of time until the event starts and then send the notification to the user. There are also some conditions in place to keep the app from making the app from making repeated processes from the same user.

Database

Every table used in the application was created using Django's ORM feature. A class is created in the models.py file of each Django app. In this class, the database table columns and types are specified. Once done, the commands "python manage.py makemigrations *appname*" and "python manage.py migrate" are executed to create the tables.

Deployment

The application was deployed to Heroku. The application code was pushed to a repository found in a Heroku virtual machine. PostgreSQL was also installed in this virtual machine. The command "heroku run python manage.py migrate" was used to create the Django tables in the new database. Finally, Heroku provides an url that can be used to access the application: jikancontrol.herokuapp.com

Performance

The application's performance is very good. There are no big SQL functions or stored procedures that require a lot of time to execute. Also, the machine learning models are stored with the server side code. This means that the application has quick access to the prediction results.

Projected Costs

The adoption of this application is expected to be slow. For that reason it seems like a good option to keep the application hosted on Heroku. However, in order to improve the performance and the load capacity, a new hosting pricing plan would need to be chosen. To get started, the “Production” plan and the standard PostgreSQL plans would be chosen. These cost \$50 a month each and provide up to 1000 connections. As more people start to use the application, then the “Premium” plans are the next choice. The container plan has a price of \$200/month, while the PostgreSQL plan’s price is \$750/month and provides up to 5000 connections. Finally, once the number of users starts nearing 5000, a “Private” plan would be chosen and the prices would have to be negotiated. The projected cost of hosting the application is between \$100 to \$1000 per month during the initial and intermediate period.

Testing

The UI and what can be done in it was tested using Selenium WebDriver. Selenium is an automated testing tool which simulates how a human interacts with the application. Python code is created in order to indicate where the webdriver will navigate, click, and send text to.

Pros and Cons

The biggest downside of the application is that the age of the samples in the dataset doesn't fully match the age of the target users. This is important because the effort that the users put in when preparing for an exam can be greater than the effort put by the people in the dataset. On the other hand, as more people use the application, better data will be gathered that can be used to produce better predictions.

Lessons Learned

The most important lesson learned was that no matter how simple an application looks, it is very difficult to estimate the actual work that went and currently goes into it. Calendars look extremely simple. Simple enough that any programmer can make one. However, that was not the case in this project because it was the most complex part. Another lesson is that not too much data is needed to create something useful. The dataset used in this project was on the smaller size, however it was still able to yield good results.

Future Work

This application doesn't have to be limited to just predicting exam scores. It can also be used to keep track of other relevant things in the user's life. For example, the calendar would be perfect for keeping track of credit card payments. Also, a machine learning model could be created that can predict if a user will default in their payments by analyzing their payment history. There could also be the possibility of adding the ability to sync with a student's courses to have deadlines pop up automatically.

Applied Standards

ISO/IEC 9126 was the software development standard used to ensure the quality of the application.

Functionality

- a) Suitability - When Jikan Control was first defined, the main functionalities that were promised were the ability to create events, exams, notes, and the ability to predict exam scores. The functions implemented in the Calendar and in the Detail pages provide these proposed functionalities.
- b) Accuracy - The application provides a Calendar that users use to keep track of their time. It also provides exam score predictions which users use to modify the way and how much time they study. Both are the main reasons for the creation of this application.

- c) Interoperability - The application is able to interact with Gmail. It uses an account to send automated password recovery emails to users that request them.
- d) Security - Access to events, notes, exams, profiles, and calendars are only available to the user that created or owns them. Server side logic was implemented to prevent unauthorized access to any entity or page that belongs to another user.
- e) Functionality compliance - Jikan Control does not store any sensitive data that could potentially harm the users if exposed. It also doesn't host any copyrighted material.

Reliability

- a) Maturity - Since this application is hosted on the Heroku platform, the company provides easy access to an error log to determine what went wrong when there is an error in the application.
- b) Fault tolerance - The hosting plans used in the Heroku platform will allow for a capacity that exceeds the current number of users. Since this application will be first adopted by students of UPR Mayaguez then the load produced by the users won't be enough to crash the system.
- c) Recoverability - Heroku provides the ability to roll back up to 7 days (or more if a private plan is chosen). The company also provides "Continuous Protection on Heroku Postgres" which provides logs that can recover databases to a state before failure.

Usability

- a) Understandability - Most users will already be familiar with calendars such as the Gmail or Outlook Calendars. The way that the Jikan Calendar is used is very similar to those applications so the users will quickly understand what they are seeing.
- b) Learnability - Learning to use the application means navigating with a side menu, submitting and viewing forms. Once a user creates their first event, exam or note, they will have learned how to interact with the majority of the system.
- c) Operability - The Heroku platform ensures that 15 minutes is the maximum amount of downtime that the application can suffer from.
- d) Attractiveness - The User Interface works well with any device that accesses the application.

Efficiency

- a) Time behaviour - There is a small amount of data transfer between the frontend and the backend system, so performance is fast. This also applies to the use of the machine learning models. Since these are stored in the same server as the backend code, then access to prediction results is fast.
- b) Resource utilization - A small number of resources are used due to the small amount of data that is being transferred.

Maintainability

- a) Analyzability - The error logs provided by the Heroku platform are used to analyze what went wrong and then are used to replicate the issues in a local environment in order to find a solution.
- b) Changeability - Changes in the application are easy unless they involve potential data loss. Database schema changes are done automatically when deploying to the Heroku Platform.
- c) Stability - Since the application does not store sensitive data, user data can be used to test the system's stability in a local environment before deploying changes to the production environment.
- d) Testability - Similar to Stability, using user data testing the application is made easy. In addition, automated tests with Selenium Webdriver were created which make testing easier and faster.

Portability

- a) Adaptability - Jikan Control is a web application with an UI that was designed to be responsive with multiple screen sizes. In other words, when accessed by most devices, the application adapts appropriately.
- b) Installability - There is no need to install the application in another environment since it can be accessed by any device.

- c) Co-existence - The application does not interfere with any existing application except when it creates a desktop push notification. However this also doesn't interfere with other running programs.
- d) Replaceability - Most of the application can be replaced as long as the new system or system structure is compatible with the existing user data (events, exams, notes, etc).

Links for Jikan Control

The following are the relevant links for this project:

- Source Code: https://github.com/axviel/JIKAN_CONTROL
- Live Production Version: <https://jikancontrol.herokuapp.com/>
- Project Page: https://axviel.github.io/JIKAN_CONTROL/
- PowerPoint Presentation:
https://github.com/axviel/JIKAN_CONTROL/wiki/files/Jikan_Control_Presentation.pptx
- Manual: https://github.com/axviel/JIKAN_CONTROL/wiki/Manual
- Gantt Chart:
https://github.com/axviel/JIKAN_CONTROL/wiki/files/Final_Report_Gantt_Chart.pdf

References

1. BMC Blogs. 2018. *Mean Squared Error & R2 Score Clearly Explained*. [online] Available at:
<https://www.bmc.com/blogs/mean-squared-error-r2-and-variance-in-regression-analysis/> [Accessed 10 May 2020].
2. Bootstrapious - Free Bootstrap Themes & Templates. 2018. *Bootstrap Sidebar Tutorial - Step-By-Step Tutorial With 5 Sidebar Templates [Updated In 2020]*. [online] Available at:
<https://bootstrapious.com/p/bootstrap-sidebar> [Accessed 10 May 2020].
3. Darman, H., Musa, S., Ramasamy, R. and Rajeswari, R., 2019. *Predicting Students' Final Grade In Mathematics Module Using Multiple Linear Regression*. 7th ed. [ebook] IJRTE. Available at:
<https://www.ijrte.org/wp-content/uploads/papers/v7i5s/ES2162017519.pdf> [Accessed 17 May 2020].
4. Docs.djangoproject.com. 2020. *Built-In Template Tags And Filters | Django Documentation | Django*. [online] Available at:
https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatefilter-json_script [Accessed 29 March 2020].
5. Docs.djangoproject.com. 2020. *Django documentation | Django*. [online] Available at:
<https://docs.djangoproject.com/en/3.0/> [Accessed 22 Feb. 2020].
6. Docs.djangoproject.com. 2020. *Widgets | Django Documentation | Django*. [online] Available at: <https://docs.djangoproject.com/en/3.0/ref/forms/widgets/> [Accessed 29 March 2020].

7. Docs.python.org. 2020. *Multiprocessing — Process-Based Parallelism — Python 3.8.3Rc1 Documentation*. [online] Available at:
<<https://docs.python.org/3.8/library/multiprocessing.html>> [Accessed 10 May 2020].
8. Docs.python.org. 2020. *Threading — Thread-Based Parallelism — Python 3.8.3Rc1 Documentation*. [online] Available at: <<https://docs.python.org/3.8/library/threading.html>> [Accessed 10 May 2020].
9. En.wikipedia.org. 2020. *ISO/IEC 9126*. [online] Available at:
<https://en.wikipedia.org/wiki/ISO/IEC_9126> [Accessed 16 May 2020].
10. Fontawesome.com. 2020. *Font Awesome*. [online] Available at:
<<https://fontawesome.com/>> [Accessed 10 May 2020].
11. Freitas, V., 2020. *Django Tips #1 Redirect*. [online] Simple is Better Than Complex. Available at:
<https://simpleisbetterthancomplex.com/tips/2016/05/05/django-tip-1-redirect.html>
[Accessed 29 March 2020].
12. Gaunt, M., 2020. *Service Workers: An Introduction | Web Fundamentals*. [online] Google Developers. Available at:
<<https://developers.google.com/web/fundamentals/primers/service-workers>> [Accessed 10 May 2020].
13. GitHub. 2020. *Safwanrahman/Django-Webpush*. [online] Available at:
<<https://github.com/safwanrahman/django-webpush>> [Accessed 10 May 2020].
14. Herman, M., 2019. *Dockerizing Django With Postgres, Gunicorn, And Nginx*. [online] Testdriven.io. Available at:
<<https://testdriven.io/blog/dockerizing-django-with-postgres-gunicorn-and-nginx/>>
[Accessed 10 May 2020].

15. Howchoo.com. n.d. *Working With Datetime Objects And Timezones In Python*. [online]
Available at:
<<https://howchoo.com/g/ywi5m2vkodk/working-with-datetime-objects-and-timezones-in-python>> [Accessed 10 May 2020].
16. Ihler, A., 2013. *Linear Regression (5): Bias And Variance*. [video] Available at:
<<https://www.youtube.com/watch?v=djRh0Rkqygw>> [Accessed 10 May 2020].
17. Kaggle.com. 2020. *Learn Python, Data Viz, Pandas & More | Tutorials | Kaggle*. [online]
Available at: <https://www.kaggle.com/learn/overview> [Accessed 22 Feb. 2020].
18. Neat Developer, 2018. *Speed Coding Calendar - JS, HTML, CSS*. [video] Available at:
<<https://youtu.be/kAXwOHh50LU>> [Accessed 10 May 2020].
19. Postgresql.org. 2020. *PostgreSQL: Documentation*. [online] Available at:
<https://www.postgresql.org/docs/> [Accessed 22 Feb. 2020].
20. Scikit-learn.org. 2020. *API Reference — Scikit-Learn 0.22.2 Documentation*. [online]
Available at:
https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model
[Accessed 29 March 2020].
21. Shin, T. 2020. *All Machine Learning Models Explained in 6 Minutes*. [online] Medium.
Available at:
<https://towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a> [Accessed 22 Feb. 2020].
22. Singh Chauhan, N., 2020. *A Beginner'S Guide To Linear Regression In Python With Scikit-Learn*. [online] Medium. Available at:
<https://towardsdatascience.com/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-83a8f7ae2b4f> [Accessed 29 March 2020].

23. Stack Exchange. 2016. *Explaining The Variance Of A Regression Model*. [online]

Available at:

<<https://stats.stackexchange.com/questions/203540/explaining-the-variance-of-a-regression-model>> [Accessed 10 May 2020].

24. Student Alcohol Consumption. 2020. [Dataset]

<https://www.kaggle.com/uciml/student-alcohol-consumption>.