T.C.

ISTANBUL AYDIN UNIVERSITY

FACULTY OF ENGINEERING

GPA PREDICTION SYSTEM

GRADUATION THESIS

Emre ERDİN

Department : Engineering Department

Program : Software Engineering

December, 2022

T.C.

ISTANBUL AYDIN UNIVERSITY

FACULTY OF ENGINEERING

GPA PREDICTION SYSTEM

GRADUATION THESIS

Emre ERDİN

(B1805.090031)

Department : Engineering Department

Program : Software Engineering

Supervisor : Prof. Dr. Ali OKATAN

December, 2022

DECLARATION

We hereby declare with respect that the study “GPA Prediction System”, which we submitted as a graduation project thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the project phase to the conclusion of the thesis and that the works we have benefited are from those shown in the Bibliography. (…/…/20...)

PREFACE

GPA Prediction System project has been developed under the supervision of Prof. Dr. Ali Okatan who has been supportive and a knowledge source throughout the project development process.

December 2022 **Emre ERDİN**

ABSTARACT(ENGLISH)

GPA calculation is done by having different scores of multiple classes taken by students, multiplying them by a coefficient originated by weights of classes, and dividing this result by an average value. In this calculation, different methodologies are used. One of these methodologies is "*Letter Grading*". In this method, each letter refers to a score where A is equal to 4 and all the rest letter grading depends on this criterion. The purpose of "*GPA Prediction System*" project is to predict grading scores of the students out of 4 depending on the life standards and habits that they have. In GPA Prediction System, a data set collected by the students is used as a training and test set to train the machine learning model and unknown data prediction is done by using this trained machine learning model. What GPA Prediction System aims is to prove that in which conditions and standards the GPA of a student can change and may be.

ÖZET(TURKISH)

GNO hesabı, öğrencilerin birden fazla dersten aldıkları farklı puanların sınıf ağırlıklarından elde edilen bir katsayı ile çarpılması ve elde edilen sonucun ortalama bir değere bölünmesi ile yapılır. Bu hesaplamada farklı metodolojiler kullanılmaktadır. Bu metodolojilerden biri de “Harf Notlandırma”dır. Bu yöntemde, her harf, A'nın 4'e eşit olduğu bir puanı ifade eder ve geri kalan tüm harf notları bu kritere bağlıdır. “GPA Tahmin Sistemi” projesinin amacı, öğrencilerin sahip oldukları yaşam standartları ve alışkanlıklarına göre 4 üzerinden notlarını tahmin etmektir. GPA Tahmin Sisteminde, öğrenciler tarafından toplanan bir veri seti, makine öğrenmesi modelini eğitmek için eğitim ve test seti olarak kullanılır ve bu eğitilmiş makine öğrenmesi modeli kullanılarak bilinmeyen veri tahmini yapılır. GPA Tahmin Sisteminin amacı, bir öğrencinin GPA'sının hangi koşul ve standartlarda değişebileceğini ve olabileceğini kanıtlamaktır.

TABLE OF CONTENTS

DECLARATION ..... **ii**

PREFACE ..... **iii**

ABSTARACT(ENGLISH) ..... **iv**

ÖZET(TURKISH) ..... **v**

TABLE OF CONTENTS ..... **vi**

I. INTRODUCTION **1**

A. What Is GPA? 1

B. What is the Purpose of GPA Prediction System? 1

C. How It Could Be Helpful? 1

II. METHOD **2**

A. How Does GPA Prediction System Work? 2

B. How Was Data Set Created? 2

C. Which Type of Data Was Needed? 2

D. Machine Learning 3

1. Supervised Learning *3*

E. Which Method Is Used? 4

F. Ethics 4

III. PROJECT **5**

A. Main Purpose of Project 5

B. How We Process Data ? 6

C. Development Process 7

1. Planning *7*

2. Design *7*

3. Implementation *7*

4. Testing *8*

5. Maintenance *8*

D. Coding And Implementation 8

1. Data Collection and Preprocessing *8*

2. Machine Learning Model *9*

3. Integration with Unity *9*

4. Testing and Validation *9*

E. Screenshots 23

F. Conclusion 27

IV. RESULTS 28

V. DISCUSSIONS 28

A. Which University Majors Cannot Use This System? 28

B. Usage Areas of GPA Prediction System 29

C. Limitations and Implications for Feature Research 29

VI. REFERENCES 29

VII. APPENDIX 30

I. INTRODUCTION

A. What Is GPA?

GPA calculation is done by having different scores of multiple classes taken by students, multiplying them by a coefficient originated by weights of classes, and dividing this result by an average value. In this calculation, different methodologies are used. One of these methodologies is "*Letter Grading*". In this method, each letter refers to a score where A is equal to 4 and all the rest letter grading depends on this criterion.

B. What is the Purpose of GPA Prediction System?

Main purpose of "*GPA Prediction System*" is to prove that impacts of different criterions such as time spending for traveling, studying and personal habits, different life conditions for students for their studying standards on students' GPA score.

C. How It Could Be Helpful?

GPA Prediction System can be helpful in a way that students can see their performance evaluation based on their current studying criterion, time management and stress level. After completing the survey, they can achieve different GPA results that can show that which factor affects this GPA change more than the others. Shortly, it helps students to change their habits, time management methodologies and studying strategies to increase their GPA.

II. METHOD

A. How Does GPA Prediction System Work?

GPA Prediction System is simply a machine learning project in which we use Multiple Linear Regression algorithm that is under root of Supervised Learning methodologies. Multiple Linear Regression is used since the system tries to predict a continuous variable depending on the feature vector or independent variables that are taken by the user of system. After having a user who takes the survey, the system creates a feature vector for this survey result and depending on the model it generates by using the previous data set that was also taken by people by applying a survey. The system creates a machine learning model, trains itself and tests its model. Then calculates its *Mean Squared Error* to find the error between the real result and the predicted result. Then depending on this difference, the model processes newly coming data to predict the result of a student.

B. How Was Data Set Created?

Data set was generated by using a survey system. In the system, 10 questions were defined that were thought as base criterions for evaluation of a student's GPA result. The survey had been published via Internet and announced in different platforms to reach students. After collecting the survey results, the conversion of this data set into a feature vector was done by using Python programming language in *.csv* form.

C. Which Type of Data Was Needed?

In order to determine data type or information type, small research in a small group of students was made. It was aimed that what criterions had higher priority than others and questions in the survey were decided based on this research. The questions are as follow:

1- Gender
2- Do you repeat your classes daily?
3- For your classes, do you work alone or as a group of friends?
4- While studying, do you prefer reading or writing?
5- How many hours, on average, do you spend for your daily transportation?

6- How many hours do you study in your exam week?
7- How many points would you give for your stress level in your exam period?
8- How many hours do you sleep in your exam period?
9- How many points would you give for caffeine consumption in your exam period?
10- What is your GPA?

D. Machine Learning

Machine Learning is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, agriculture, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers, but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the working of a biological brain. In its application across business problems, machine learning is also referred to as predictive analytics.

1. Supervised Learning

Supervised learning is a machine learning paradigm for problems where the available data consists of labelled examples, meaning that each data point contains features (covariates) and an associated label. The goal of supervised learning algorithms is learning a function that maps feature vectors (inputs) to labels (output), based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is

a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias). This statistical quality of an algorithm is measured through the so-called generalization error.

E. Which Method Is Used?

GPA Prediction System project is based on a machine learning model called Multiple Linear Regression in which a labeled data set is used. Previously collected data is arranged as a feature vector to train the machine learning model. After training and testing the model, we calculate the mean squared error of the actual values and predicted values. After error calculation, till the error decreases till to the minimum value, the model continues its training by using the data set that was splatted as test set and training set.

F. Ethics

Throughout the development and research process, the most ethical issue was to get information from people. Since the questions were about their personal life, the process had to be managed carefully. Therefore, the questions were prepared in a way that none of personal information was collected. All information collected by people was just a regular or simple questions about their habits. For any ethical or personal information leaking issue, the data set was collected by using a website which does not show any device information. The data set was allowed to contain only answers of the questions.

III. PROJECT

The GPA Prediction System is a machine learning project that uses Multiple Linear Regression algorithm to predict the GPA of a student based on a feature vector or independent variables. The system creates a machine learning model by training and testing itself with previous survey results from students. The Mean Squared Error is calculated to determine the accuracy of the model's predictions. The system can then use this model to predict the GPA of new students. Overall, the GPA Prediction System is a useful tool for predicting a student's GPA, but its accuracy depends on the quality of the data used to train the model.
Considering all these explanations, the methods we followed for our project and used while creating the coding part are as follows.

A. Main Purpose of Project

The GPA Prediction System is a machine learning project with the primary purpose of predicting a student's GPA based on various factors and criteria. One of the primary goals of the system is to identify the impact of different criteria on a student's GPA, such as their study habits, time spent on traveling, personal habits, and various life conditions. The system works by using multiple linear regression algorithms to predict a continuous variable based on a feature vector or independent variables. When a student takes the survey, the system creates a feature vector for that survey result, and based on the model generated from previous data sets, it predicts the student's GPA. The system then trains and tests itself using these previous data sets, calculating the Mean Squared Error to determine the accuracy of the model's predictions. The GPA Prediction System's main goal is to help students and educators better understand the factors that impact a student's GPA, allowing for better academic planning and support. By analyzing the various criteria that affect a student's GPA, the system can help identify areas where students may need additional support or resources to improve their grades. This could include tutoring or study groups, accommodations for travel or personal needs, or even adjustments to class schedules or academic requirements. Overall, the GPA Prediction System is an innovative tool that can provide valuable insights into a student's academic performance and help identify areas for improvement. By using machine learning

algorithms to predict GPA, the system can help educators and students alike better understand the complex factors that contribute to academic success, ultimately leading to better outcomes for all involved.

B. How We Process Data ?

In the GPA Prediction System project, the data processing starts with creating a survey system to collect data from students. The survey consists of 10 questions that were determined based on the research conducted on a small group of students. The questions cover different aspects that are thought to have an impact on students' GPA results, such as studying habits, stress level, caffeine consumption, daily transportation time, and sleep hours during exam periods. The collected survey data is then converted into a feature vector using Python programming language and saved in .csv format. The feature vector is used as input to train a machine learning model based on Multiple Linear Regression, which is a supervised learning algorithm. In supervised learning, the available data consists of labeled examples, meaning that each data point contains features and an associated label. In this case, the label is the student's GPA score. The goal of the machine learning model is to learn a function that maps the feature vector to the GPA score, based on example input-output pairs from the training data. The training process involves splitting the data set into training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate its performance. The model is trained by minimizing the mean squared error between the predicted GPA score and the actual GPA score from the training data set. This process continues until the error decreases to a minimum value, indicating that the model has learned the underlying patterns in the data set. Once the model is trained, it can be used to predict the GPA score of a new student based on their feature vector. This prediction can be made by feeding the feature vector into the machine learning model and obtaining the predicted GPA score as output. The accuracy of the prediction can be evaluated by comparing it to the actual GPA score of the student. In summary, the data processing in the GPA Prediction System project involves collecting survey data from students, converting it into a feature vector, and using it to train a machine learning model based on Multiple Linear Regression. The model is trained by minimizing the mean squared error between the predicted GPA score and the actual

GPA score from the training data set. Once the model is trained, it can be used to predict the GPA score of a new student based on their feature vector.

C. Development Process

The development process of a project involves several stages, including planning, design, implementation, testing, and maintenance. Each stage requires careful consideration to ensure the final product meets the desired requirements and objectives. In this case, we will explore the development process of a project that uses Unity and Python libraries.

1. Planning

During the planning stage, the objectives and requirements of the project are identified. This includes defining the purpose of the project, outlining the features and functionality required, and establishing a timeline for completion. For this project, the objective is to create a GPA Prediction System using Unity and Python libraries to predict a student's GPA based on various factors.

2. Design

Once the planning stage is complete, the design stage begins. This involves creating a blueprint of the project, including its architecture, features, and functionality. For this project, the design involves creating a user interface in Unity that allows users to input data, such as their gender, study habits, and GPA, and then using Python libraries to predict the student's GPA based on the input data.

3. Implementation

During the implementation stage, the design is brought to life through coding. In this case, Unity is used to create the user interface, while Python libraries are used to process the data and make the GPA predictions. This involves coding the algorithms, scripts, and other necessary components required to make the project work as intended.

4. Testing

The testing stage involves ensuring the project works correctly and meets the desired objectives. In this case, the project is tested to ensure that the user interface works as intended, the data is processed correctly, and the GPA predictions are accurate. Any issues or bugs that are identified during testing are addressed and fixed before the project is released.

5. Maintenance

Once the project is released, maintenance is required to ensure it continues to work correctly and remains up to date. This involves updating the project with new features or functionality, addressing any issues that arise, and ensuring it remains compatible with any new technology or software updates.

In summary, the development process of a project using Unity and Python libraries involves several stages, including planning, design, implementation, testing, and maintenance. Each stage requires careful consideration to ensure the final product meets the desired objectives and functions correctly.

D. Coding And Implementation

The GPA Prediction System is a machine learning project developed using Python and Unity. It predicts a student's GPA based on various criteria such as gender, study habits, stress levels, and daily transportation, using the Multiple Linear Regression algorithm.

1. Data Collection and Preprocessing

The first step in the project was to collect data from students through an online survey. The survey was designed to gather information about the students' habits, study methods, and other factors that might affect their GPA. The data was then preprocessed and converted into a feature vector using Python's Pandas library. The feature vector included data such as gender, study habits, stress levels, caffeine consumption, and transportation time.

2. Machine Learning Model

Once the data was collected and preprocessed, a machine learning model was built to predict the students' GPA based on the feature vector. Multiple Linear Regression model was chosen to train the model. The model was implemented in Python using machine learning libraries such as Scikit-Learn and TensorFlow.

3. Integration with Unity

The next step was to integrate the machine learning model with Unity. The project was developed in Unity Game Engine, which provides a user-friendly interface for building games and applications. To integrate the model with Unity, a Python script was created that could be called from within Unity. The script read the data from a .csv file, passed it through the trained machine learning model, and returned the predicted GPA value. The script was then integrated with Unity using the Unity's C# scripting language.

4. Testing and Validation

Once the implementation was completed, the project was tested and validated to ensure that it was working as intended. The model's performance was evaluated using metrics such as mean squared error and accuracy. The project was also tested with different input values to ensure that it was providing accurate predictions.

Overall, the development process of the GPA Prediction System project involved collecting and preprocessing data, building and training a machine learning model, integrating the model with Unity using Python libraries, and testing and validating the project. This project demonstrates the power of machine learning and how it can be integrated with game engines to create innovative applications.

5. Source Code

There are 8 code files in total in the project. 7 of these files were developed in C# and the last 1 in Python coding languages which does the prediction part.
First one of our scripts named ***Answer.cs.*** The mentioned script is like that:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Answer : MonoBehaviour
{
    public int answer1;
    public int answer2;
    public int answer3;
    public int answer4;
    public int answer5;
    public int answer6;
    public int answer7;
    public int answer8;
    public int answer9;
    public static Answer answers;


    public List<GameObject> questions;
    public int questionIndex;

    public bool end;
    // Start is called before the first frame update
    void Start()
    {
        answers = this;

        questionIndex = 0;
        for (int i = 0; i < questions.Count; i++)
        {
            questions[i].gameObject.SetActive(false);
        }
        questions[questionIndex].gameObject.SetActive(true);
    }
```

```
    // Update is called once per frame
    void Update()
    {

    }
}
```

This C# script is used in Unity game engine to manage a set of questions and answers for the user to interact with. The script defines a public class named "*Answer*" that inherits from *MonoBehaviour* class. The script has nine public integer variables that are intended to store the answers to questions, a static variable for global access, a list of *GameObjects* representing the questions, an integer variable to track which question the user is on, and a boolean variable to signal the end of the quiz. In the Start method, the script initializes the questions list and sets all question objects to be inactive except for the first one. The Update method is currently empty. The second script of project is ***AnswerButton.cs***. The named script is like that:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class AnswerButton : MonoBehaviour
{
    public int answerValue;
    private Slider parentSlider;
    private Text sliderText;
    // Start is called before the first frame update
    void Start()
    {
        if (transform.parent.GetComponent<Slider>())
        {
            parentSlider = transform.parent.GetComponent<Slider>();
```

```
            sliderText =
GameObject.FindGameObjectWithTag("SliderValue").GetComponent<Text>();
        }
    }

    // Update is called once per frame
    void Update()
    {

    }

    public void SetAnswer1()
    {
        Answer.answers.answer1 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer2()
    {
        Answer.answers.answer2 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer3()
```

```
    {
        Answer.answers.answer3 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer4()
    {
        Answer.answers.answer4 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer5()
    {
        Answer.answers.answer5 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer6()
    {
```

```
        Answer.answers.answer6 = int.Parse(parentSlider.value.ToString());

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer7()
    {
        Answer.answers.answer7 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer8()
    {
        Answer.answers.answer8 = answerValue;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
    }
    public void SetAnswer9()
    {
        Answer.answers.answer9 = answerValue;
```

```
Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(fa
lse);
        Answer.answers.questionIndex++;
        if (!(Answer.answers.questions.Count <= Answer.answers.questionIndex)) {

Answer.answers.questions[Answer.answers.questionIndex].gameObject.SetActive(tr
ue);
        }
        else
        {
            Answer.answers.end = true;
            Controller.controller.surveyQuestionPanelUI.SetActive(false);
            Controller.controller.loadingVideo.SetActive(true);
        }

    }

    public void SetSliderText()
    {
        sliderText.text = parentSlider.value.ToString();
    }
}
```

The code defines a class called *AnswerButton* that manages the behavior of answer buttons in a survey. It has a public integer variable called *answerValue*, and private Slider and Text variables called *parentSlider* and *sliderText* respectively. In the *Start()* function, if the parent object of the answer button has a Slider component, it gets assigned to the *parentSlider* variable and the *sliderText* variable is assigned to a Text object with the tag "*SliderValue*". There are nine public functions called *SetAnswer1(), SetAnswer2(), SetAnswer3(),* etc., each of which sets a different answer variable in a class called Answer, which is presumably being used to store the answers to the survey questions. After setting the answer, the function deactivates the current question and activates the next one in the sequence. The *SetAnswer6()*

function is slightly different, as it gets the value of the *parentSlider* and sets the *answer6* variable to that value after converting it to an integer. The *SetAnswer9()* function is also different, as it checks whether there are any more questions left in the survey. If there are, it deactivates the current question and activates the next one. If not, it sets a variable called end to true, deactivates the survey question panel UI, and activates a loading video. Finally, there is a function called *SetSliderText()* that sets the text of the *sliderText* variable to the value of the *parentSlider* variable.
The third script of project is ***Controller.cs.*** The mentioned script is like that:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Controller : MonoBehaviour
{
   public GameObject mainMenuUIPanel;
   public GameObject surveyQuestionPanelUI;
   public GameObject loadingVideo;
   public GameObject resultPanel;
   public static Controller controller;
   // Start is called before the first frame update
   private void Awake()
   {
      Application.targetFrameRate = 60;
   }
   void Start()
   {
      controller = this;
      mainMenuUIPanel.SetActive(true);
      surveyQuestionPanelUI.SetActive(false);
      loadingVideo.SetActive(false);
      resultPanel.SetActive(false);
   }
```

```
    // Update is called once per frame
    void Update()
    {

    }

    public void StartSurvey()
    {
        mainMenuUIPanel.SetActive(false);
        surveyQuestionPanelUI.SetActive(true);

    }
}
```

The code manages different UI panels in a survey application. It has public *GameObject* variables called *mainMenuUIPanel, surveyQuestionPanelUI, loadingVideo*, and *resultPanel*, each of which corresponds to a different panel in the application. In the *Awake()* function, the target frame rate of the application is set to 60. In the *Start()* function, a static reference to the Controller object is created, and the different UI panels are initially set to active or inactive. Finally, there is a public function called *StartSurvey()* that sets the *mainMenuUIPanel* to inactive and the *surveyQuestionPanelUI* to active, effectively starting the survey.
The fourth script in project Is ***LoadingEnd.cs***. The named cs script is under below:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LoadingEnd : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
```

```
    }

    // Update is called once per frame
    void Update()
    {

    }

    public void LoadingEndScreen()
    {
        Controller.controller.resultPanel.SetActive(true);
        Controller.controller.loadingVideo.SetActive(false);

        MLModel.RunPythonModel();

    }
}
```

In this part of our code, a loading video is played on the screen while the guessing process is performed during cutscenes or after the answers given. This video may be longer or shorter depending on the response from predicting.
The fifth part of in the project is ***MLModel.cs.*** The mentioned script is like that:

```
using UnityEditor;
using UnityEngine;
using System.IO;
using UnityEditor.Scripting.Python;
public class MLModel : MonoBehaviour
{
    public static MLModel mlModel;
    private void OnEnable()
    {
        mlModel = this;
```

```
    }

    public static void RunPythonModel()
    {
        string scriptPath = Application.dataPath +
"/ProjectFolder/new_python_script.py";
        PythonRunner.RunFile(scriptPath);
    }
}
```

The code defines a class called *MLModel* that runs a *Python* script when called. It has a public static reference to the *MLModel* object and an *OnEnable()* function that sets the reference to this object. There is a public static function called *RunPythonModel()* that creates a string variable called *scriptPath* with the path to a Python script in the project folder. It then calls a function from the *UnityEditor.Scripting.Python* namespace called *PythonRunner.RunFile()* with the *scriptPath* variable as an argument, which executes the *Python* script.
The six part of the project is ***SurveyQuestionController.cs.*** The script part is like that:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SurveyQuestionController : MonoBehaviour
{
    public Animator selfAnimator;
    // Start is called before the first frame update
    void Start()
    {
        StartCoroutine(StopAnimator());
    }

    // Update is called once per frame
```

```
    void Update()
    {

    }
    public Ienumerator StopAnimator()
    {
        yield return new WaitForSeconds(1);
        selfAnimator.enabled = false;
        StopAllCoroutines();
    }
}
```

This is a simple Unity component that controls the animation of a UI element. When the component is started, it waits for one second before disabling the animator attached to the same *GameObject*, effectively stopping the animation. This behavior is achieved through a coroutine, which allows for a delayed execution of code. The script doesn't have any update logic, so it just sits idle after the animation is stopped. The seventh 'cript of the project is ***Python_Manager.cs.*** The code will shown here:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Python_Manager : MonoBehaviour
{
}
```

This script is currently empty and doesn't contain any functionality. However, while creating the project we have to use a Python code for predicting and we used this script to add Python functionality to your Unity project.
And the last and most important script of the project is Machine Learning part which is ***gpa_predict_ml.py.*** The script is like that:

```
from sklearn.preprocessing import StandardScaler
import UnityEngine as ue
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

uiObject = ue.GameObject.FindWithTag("Results")
answers = uiObject.GetComponent("Answer")
# Load dataset
df = pd.read_csv('Assets/ProjectFolder/gpaPred.csv')
df = df.dropna()

# Split the data into training and testing sets
X = df.iloc[:, :9]  # input features (first 9 columns)
y = df.iloc[:, 9]  # target variable (10th column)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

# Scale the data
# You can choose to scale the data using StandardScaler or MinMaxScaler
# Here's an example using StandardScaler:
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
```

```
print('Mean Squared Error:', mse)

# Make predictions
# Here's an example using a single record from the dataset
record = [[answers.answer1, answers.answer2, answers.answer3, answers.answer4,
        answers.answer5, answers.answer6, answers.answer7, answers.answer8,
answers.answer9]]
record_scaled = scaler.transform(record)
prediction = model.predict(record_scaled)

predictionUnityText = ue.GameObject.FindWithTag(
   "Prediction")
pred_str = str(prediction).replace('[', '').replace(']', '')
predictionUnityText.GetComponent("Text").text = pred_str
mseUnityText = ue.GameObject.FindWithTag(
   "MSE")
mse_str = str(mse).replace('[', '').replace(']', '')
mseUnityText.GetComponent("Text").text = mse_str
```

The code is a Python script that uses the *Scikit-learn* library for machine learning and the *Pandas* library for data manipulation. It reads in a *CSV* file containing data on student performance and uses it to train a linear regression model to predict a student's GPA based on various factors in asked survey, standardized test scores, and demographic information. The script also imports the "*LinearRegression*" and "*mean_squared_error*" modules from Scikit-learn for training and evaluating the model. Next, the script locates a UI object in the Unity scene with the tag "*Results"* and gets the values of the user's answers from its components. The script then loads the data from a CSV file using the *"pd.read_csv()"* function from the Pandas library and drops any rows with missing data using the "*dropna()"* method. The data is then split into training and testing sets using the "*train_test_split()"* function from Scikit-learn. After that, the data is scaled using the "*StandardScaler()"* function from Scikit-learn to standardize the feature values to a mean of 0 and a standard deviation of 1. The "*fit_transform()"* method is used to scale the training set and the "*transform()"*

method is used to scale the testing set. The script then creates a linear regression model using the "*LinearRegression()"* function from Scikit-learn and trains the model using the training set with the "*fit()"* method. Next, the script evaluates the model using the testing set and calculates the mean squared error using the "*mean_squared_error()"* function from Scikit-learn. Finally, the script makes predictions using the trained model. It creates a list containing the user's answers to the survey questions and scales the values using the previously created scaler. The scaled values are then used to make a prediction using the "*predict()"* method of the model. The predicted value is then displayed on the Unity UI using the "*GetComponent()"* method to get the text component and the "*text*" property to set the value. The mean squared error is also displayed in a similar manner.

E. Screenshots

I have some screenshots of our project. I think that these screenshots will provide clearer images and a more understandable diagnosis of how our project works.
For example we have a starting page of our mobile application(*Figure 1.1*),there are some questions about our project. We have questions on a screen that can be solved by people who want to make their own GPA estimation.(*Figure 1.2,Figure 1.3, Figure 1.4, Figure 1.5*). After that we have loading page(*Figure 1.6*) and prediction page (*Figure 1.7*).

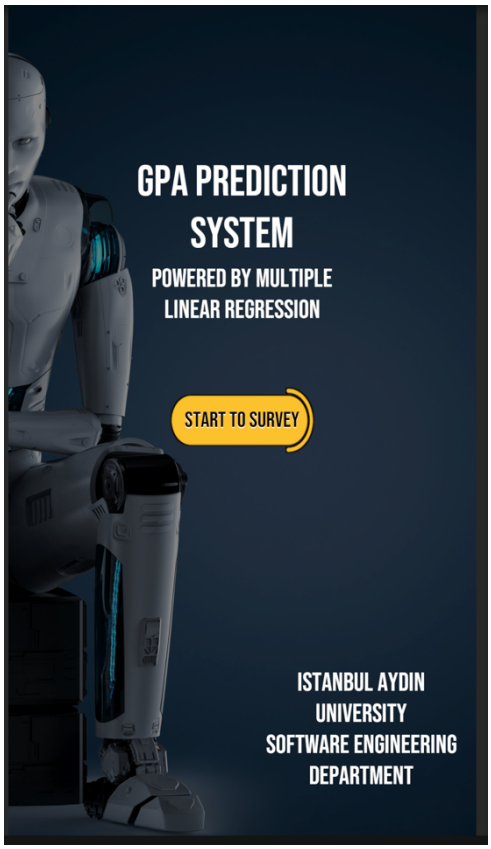

Figure 1.1

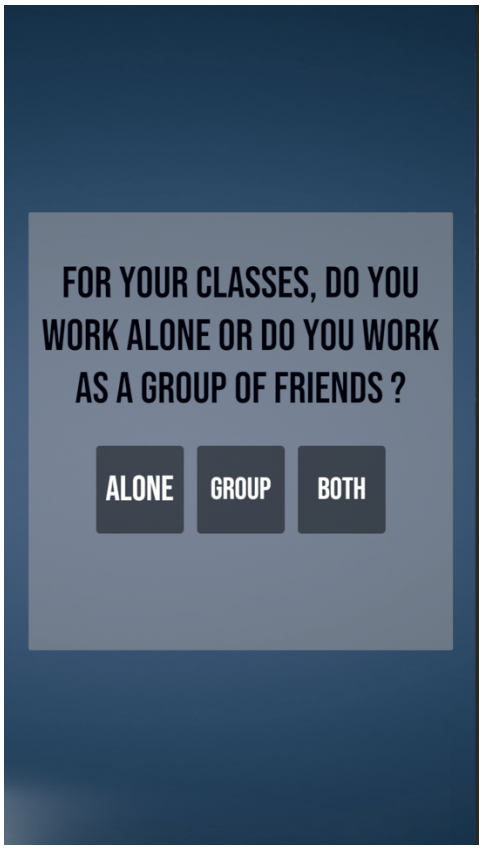

Figure 1.2

Some of the reasons for asking the specific questions in the screenshots and how they affect after asking, for example in we discussed in *Figure 1.2*, a student's academic performance may be influenced by whether they work alone or in a group. Working alone may allow for greater focus and control, while working in a group can provide opportunities for collaboration and knowledge sharing. Ultimately, the impact of working alone, in a group, or both on a student's performance may depend on individual factors such as learning style, motivation, and social skills.

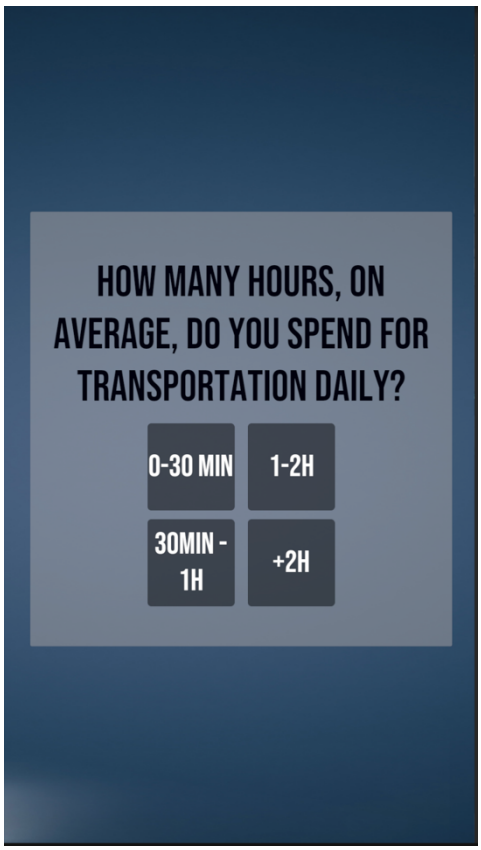

Figure 1.3

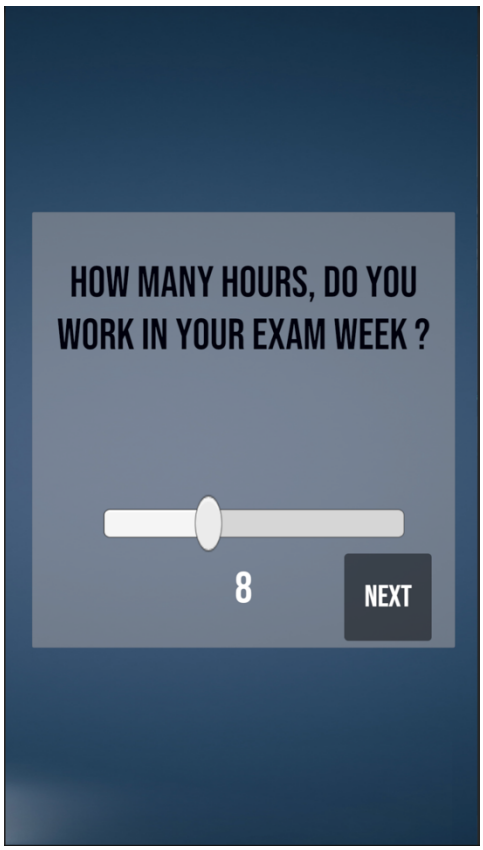

Figure 1.4

After that as asked in *Figure 1.3*, the amount of time spent on transportation daily can potentially affect a student's grades, as it may reduce the amount of time they have available to study or complete assignments. Longer commute times can also be more physically and mentally draining, which could impact a student's ability to focus and learn effectively. However, the exact degree of impact will depend on individual factors such as study habits, time management skills, and overall workload. In *Figure 1.4*, the number of hours a student spends studying for an exam can have a significant impact on their exam grade. Generally, students who study for more hours tend to perform better on exams compared to those who study for fewer hours. However, the effectiveness of the study hours can also depend on other factors, such as the quality of the study material, the student's understanding of the material, and their study habits.

As mentioned before, there are one question more which is asked in *Figure 1.5*, High levels of stress can negatively affect a student's exam performance by impairing their ability to focus and recall information. Stress can also lead to physical symptoms such as headaches and fatigue, further hindering a student's ability to perform well on exams. On the other hand, moderate levels of stress can motivate students to study

harder and perform better, but excessive stress can have the opposite effect. Therefore, managing stress levels effectively is important for students to perform well on exams. That is why we are asking these questions to students. Because all these questions are affecting too many students.

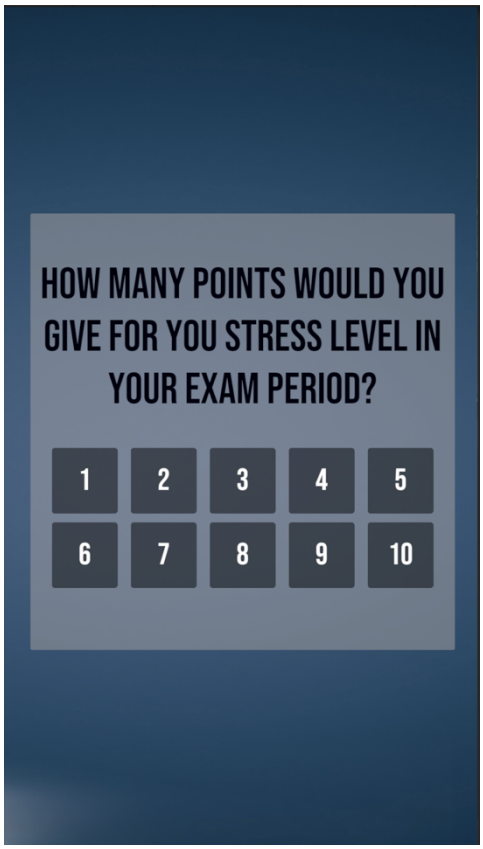

Figure 1.5

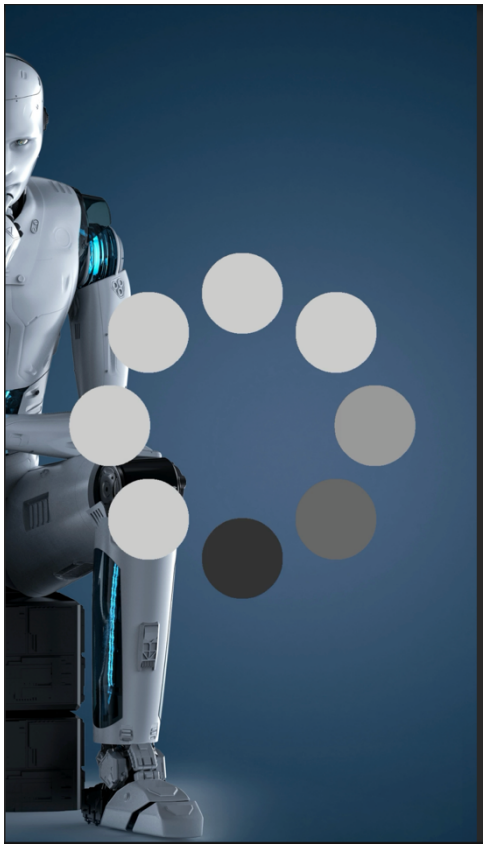

Figure 1.6

These questions are shown as examples only. After giving answers to 10 questions like this, we can learn the grade point averages of the students. Finally, we direct this calculated value to a page that we share with the students.

As can be seen, there is a loading screen in the *Figure 1.6*. This screen gives the time required for the machine learning system we use to work and calculate to get the predicted result after all the questions are answered.

Finally, we see the estimation page which is *Figure 1.7*, which is the last page of our project. we make a guess based on the answers to the questions we have on this screen. We have a deviation value for our prediction, we print it as Mean Squared Error.

GPA prediction using Multiple Linear Regression is a statistical approach where the algorithm uses a set of independent variables, such as gender, study hours, stress

level, caffeine consumption, etc. to predict a student's GPA, which is the dependent variable. The model finds the best-fit line that explains the relationship between the dependent and independent variables using a mathematical formula.

Figure 1.7

To evaluate the performance of the model, the mean squared error (MSE) is calculated. The MSE measures the average of the squared differences between the actual and predicted values. The lower the MSE value, the better the model's performance in predicting the GPA accurately. The model is trained using a labeled dataset, and after testing it on a test set, the MSE is calculated to determine the accuracy of the model.

F. Conclusion

In conclusion, the GPA Prediction System project aims to predict a student's GPA based on various factors such as their study habits, transportation time, stress levels, and caffeine consumption during exam periods. The project uses machine learning algorithms, specifically multiple linear regression, to create a model based on a labeled dataset. The model is trained and tested using the dataset, and the mean

squared error (MSE) is calculated to evaluate the accuracy of the predictions. The project highlights the importance of factors such as study habits, stress levels, and transportation time in a student's academic performance, and how these factors can be used to predict their GPA. Overall, the GPA Prediction System can be a useful tool for students and educators alike in understanding and improving academic performance.

IV. RESULTS

In this project, which aims to organize organisms, we aimed for students to achieve more successful results in their stress levels and working lives with the points they went to. We want to provide a healthier education life with the savings they have gained from these goals. The methods we used in this project were chosen accordingly. While collecting the data, while answering the questions, these suggestions are grouped in order to produce solutions and an estimation system is emphasized based on the answers given to these questions.

V. DISCUSSIONS

A. Which University Majors Cannot Use This System?

Since Grade Point Average system for some specific majors such as Faculty of Art, Faculty of Architecture cannot use this GPA Prediction System since they use different criterion for student evaluation. For instance, for an Art student, studying hours may be an important factor yet it also depends on the skills the student has. Therefore, GPA Prediction System criterions will not be sufficient to predict the result of this student. Also, for some majors of which students need physical effort, caffeine consumption may be another significant criterion.

B. Usage Areas of GPA Prediction System

After that, it can be given according to the education of the students according to the working hours and on the other questions we have taken into consideration, based on the studies that the students have done throughout the year, perhaps without the need for any exams - at a level where the project is very developed. (At least an average can be found.) At the same time, external factors will be ignored on behalf of students based on this grading system. A student who has studied for 10 hours until the last day can be prevented from getting a bad result in the exam because he is ill during the exam. It will also cause students to consume too much caffeine while working and balance their study patterns. If a student needs sleep, the program will let him know and guide him so that he can avoid caffeine consumption and get him a higher grade. In addition, it can be found out whether the students' effort for a lesson is sufficient or not. For example, an idea can be given to a student about whether he/she is working adequately based on the values he/she has entered. This will motivate the student and push him to work harder.

C. Limitations and Implications for Feature Research

For implications and limitations, different questions and different criterions can be implemented, included or removed. Also, by collecting a new data set depending on different questions may increase the accuracy of the prediction results. Yet, a limitation for questions in terms of personal information can be inspected.

VI. REFERENCES

Medium. *A Beginner's Guide to Machine Learning.* Access: Jan 23, 2018. https://medium.com/@randylaosat/a-beginners-guide-to-machine-learning-dfadc19f6caf

Medium. Supervised Learning. Access: Jun 2, 2018. https://medium.com/@jorgesleonel/supervised-learning-c16823b00c13

VII. APPENDIX

Cinsiyet (Gender)	Günlük ders tekrarı yapıyor musunuz ? (Do you repeat your classes daily ?)	Derslere arkadaş grubu ile mi çalışıyorsunuz yalnız mı çalışıyorsunuz? (For your classes do you work alone or do you work as a group of friends ?)	Çalışırken yazarak çalışmayı mı yoksa okuyarak çalışmayı mı tercih ediyorsunuz ? (While studying do you prefer reading or writing ?)	Günlük ulaşıma ortalama ne kadar zaman harcıyorsunuz ?(How many hours on average do you spend for transportation daily?)	Hour	Sınav döneminde stres düzeyinize 0-10 arasında kaç puan verirsiniz?(How many points would you give for you stress level in your exam period?)	Sınav döneminde ortalama kaç saat uyuyorsunuz ?(How many hours do you sleep in your exam period?)	Sınav döneminde kafein tüketiminize kaç puan verirsiniz?(How many points would you give for caffeine consumption in exam period?)	Genel Not Ortalamanız ? (Örneğin 3.50)GPA ? (Eg: 3.50)
2	2	3	2	3	2	7	10	8	2.48
2	2	1	3	1	4	11	5	9	2.80
1	2	3	1	1	2	1	7	11	2.32
1	1	1	1	3	8	8	4	11	3.62
1	2	3	3	3	0	11	7	1	2.83
2	2	2	3	3	6	8	7	4	2.47
2	1	3	3	4	13	11	9	2	3.85