Predicting Academic Success: Building an Effective Student Grade Predictor Using a Random Forest Regressor

ECS 171 Machine Learning
Group 21 Project Report

Group members:
Safoorah Siddiqui, Pablo Rivera, Jessisca Frost,
Kevin Aromin, and Jonathan De Haro

Github repository:
https://github.com/UCD-ECS-171-Group-21/Group21FinalProject

I. Introduction and Background

The performance of students in academics is influenced by a wide range of factors, going beyond the usual educational inputs. Over the past few years, there has been increasing recognition among researchers and educators about the significance of taking into account non-academic factors, like social dynamics and lifestyle choices, when it comes to understanding and predicting students' grades. The social environments that students find themselves in have a significant influence on their academic journeys. Various elements, including socioeconomic status, family support, peer relationships, and community involvement, can impact a student's concentration on their studies, affecting their grades, and as discussed later on in our research paper, we will highlight the correlation of some of these social factors in predicting students' academic performance. Additionally, when it comes to lifestyle choices, the correlation between alcohol consumption and academic performance has become an increasingly worrisome topic that we will dive into. Drinking too much alcohol, for example, can have serious consequences on cognitive function, attention span, and decision-making abilities, ultimately impacting academic performance. Moreover, the way gender and social identity intertwine can impact academic performance. In light of these insights, our study aims to encapsulate a holistic understanding of the multifaceted impacts that non-academic variables exert on academic success, thereby offering a more nuanced lens through which educational achievements can be assessed and enhanced.

Developing a precise machine learning model that forecasts students' academic performance by considering a wide range of factors, such as social aspects and alcohol consumption, can bring significant advantages to different parties including educators, policymakers, and health professionals. Below are a few examples of how an accurate model of predicting a student's grades and incorporating crucial features is beneficial to each party mentioned:

A. Educators

- Can assist educators in identifying students who may be prone to academic underperformance as a result of social or lifestyle issues, and get them the help they need early on.
- By making accurate predictions, educators can implement targeted interventions to address the specific needs and challenges faced by certain students. This can include counseling or support programs tailored to their unique circumstances.

B. Policymakers

- Can assist with making education policies that are based on facts and take into account all the various factors that affect a student's success.
- Policymakers can help students who are having a hard time by giving them the tools they need if they know how social and lifestyle factors affect them.

C. Health Professionals

- Can assist in learning more about the health and well-being of students, which can help health professionals come up with health promotion methods that address mental health, substance use, and lifestyle choices.

II. Literature Review

Several research studies have investigated the application of machine learning to forecast students' academic performance using different characteristics. Research frequently encompasses variables such as attendance, study habits, socioeconomic level, and previous academic achievement while some have incorporated and heavily relied on students' actual performance and participation in the class.

Yudish Teshal Badal and Roopesh Sungkur, in their recent research, have employed a Random Forest classifier to predict student grades, as discussed in the literature on machine learning models. This classifier integrates several factors pertaining to the student profile, such as past examination outcomes and experiences on a learning platform. The variables under consideration comprise numerous aspects of student engagement, including the overall number of talks, metrics of participation, duration of assignment submission, and responses to multiple-choice questions. Their model demonstrated an impressive accuracy rate of 83%, highlighting the efficacy of Random Forest in capturing intricate relationships and forecasting academic results. In addition, their study utilized a feature selection technique, eliminating unnecessary qualities and highlighting the significance of key factors in the forecasting process.

The present paper utilizes the J48 machine learning method, which is based on the well-known C4.5 algorithm created by Ross Quinlan. This algorithm is renowned for its utilization of decision trees, providing an advanced technique for forecasting student grades. The research demonstrates an exceptional prediction accuracy rate of 99.6% when employing the J48 algorithm. The exceptional precision highlights the capacity of machine learning algorithms to predict academic success, showcasing the effectiveness of J48 in delivering nearly flawless forecasts. The application of proven algorithms such as J48 adds to the increasing amount of research that emphasizes the effectiveness of machine learning in educational settings.

Before we start constructing our predictive model, it is crucial to acknowledge the importance of feature engineering techniques that have been shown in previous research. The Random Forest classifier developed by Badal and Sungkur, as well as the J48 algorithm discussed in this study, highlight the significance of choosing appropriate features to get precise predictions. To achieve accurate model prediction, it is crucial to employ comparable feature engineering techniques to recognize and rank the most impactful factors. Moreover, our research aims to attain accuracy levels that are on par with the exemplary models mentioned before, serving as a crucial standard for our work. Therefore, it is crucial to thoroughly analyze feature selection methods and algorithmic decisions in order to guarantee that our model achieves the utmost precision in forecasting student grades.

III. Dataset Description and Exploratory Data Analysis of the Dataset

Our investigation delves into the extensive ”Student Alcohol Consumption” dataset, a valuable source of information gathered from surveys among students in math and Portuguese language courses in secondary schools. This dataset provides a wide range of 33 features that are the same in both the ”student-mat.csv” (Math course) and ”student-por.csv” datasets. It allows for a detailed examination of the various social, gender, and study-related factors that impact students' academic paths. Interestingly, there are 382 students who are part of both datasets, which provides an opportunity to compare and analyze the factors that affect academic performance in these different subjects.

During our investigation, we came across distinct data files for math and Portuguese grades, which led us to adopt a methodical approach to evaluating our models. We thoroughly analyzed predictive models on both datasets separately to understand their strengths and limitations. This helped us determine which dataset, either math or Portuguese, provided the most valuable insights. The grade values, which range from 0 to 20, were presented on a numerical scale. We made sure to maintain this continuity in our analyses, as we understand the significance of preserving the inherent numerical relationships within the grading system.

Using a combination of different types of features, we used one-hot encoding to make the dataset consistent, which helped us create accurate predictive models. Next, we used Principal Component Analysis (PCA) to simplify the dataset's dimensions, making our analysis more efficient and focused. Afterwards, we created a correlation matrix to discover complex relationships within the features, helping us grasp the underlying structure.

Throughout our investigation, we discovered a clear relationship between grades 1 and 2 with grade 3 (G3), which led us to focus our predictive models on the final grade (G3) as the main variable of interest. In addition, we decided to remove the school variable from our dataset because the vast majority of students (90%) attended Gabriel Pereira School. This made this particular feature less useful for our analyses.

IV. Proposed Methodology

Our methodology aims to tackle the task of forecasting student grades, which involves predicting a finite numerical outcome of 0-20. We know that this problem is one of linear regression, given our target variable is a numerical value, so we began by running a correlation heatmap to see how we should approach feature selection. Initially, a low correlation between all variables prompted us to cherry-pick only select variables that appeared to be moderately correlated and see how our regression models fared.

In an attempt to predict what number grade a student would receive, we took three approaches: Classification Tree, Neural Network (EDA), and K-Nearest neighbors. For our three models, we manually picked the variables with the highest correlation. Looking comparatively at the displays and resulting accuracy, all resulted poorly, ranging within 20%, or resulting in high MSE and low R2 scores. As such, we decided the best approach was to perform feature selection on our data before we progressed with a linear regression-type model. Ultimately, after research and consultation with the professor, we settled on Principal Component Analysis, which worked to select usable variables for our linear regression and reduce the dimensionality of our dataset to make more accurate predictions.

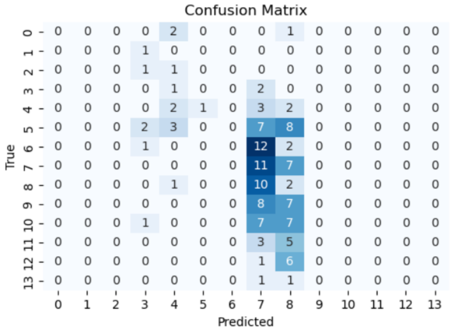
With this dataset, we tackle our grade prediction task using a Random Forest Regression model. The selection of Random Forest is based on its capacity to effectively handle non-linear associations within the data and deliver resilient predictions. Random Forests are a type of ensemble learning algorithm that combines numerous decision trees, each trained on a different subset of the data, to generate a more resilient and accurate model. The ensemble's inherent balance and diversity enhance the model's efficacy in capturing intricate interactions within the feature area.

In order to select Random Forest Regression as our chosen approach, we performed a comparative analysis that involved evaluating two alternative models: k-Nearest Neighbors (KNN) and a Neural Network. The objective was to determine the model that exhibited greater performance in forecasting student grades using the provided dataset. The performance of each model was assessed based on key metrics like Mean Squared Error (MSE).

When contemplating the implementation of our model in a real-world situation, we carried on creating an interface that would allow for user-friendly input of features. Nevertheless, as a component of our iterative methodology, we made the decision to enhance the efficiency of the user input procedure. The ultimate design entails users entering features that have undergone Principal Component Analysis (PCA) reduction. Subsequently, the diminished set of characteristics is inputted into our Random Forest Regression model. This strategy not only improves the efficiency of the application but also guarantees that the model can handle inputs that have been adjusted for retaining information and computational efficiency. Our proposed approach mainly relies on Random Forest Regression, supported by a thorough comparison with other models. The user interface we have designed focuses on PCA-reduced features, showcasing our commitment to creating a system that is both user-friendly and efficient in predicting student grades based on these identified features.

V. Experimental Results

A. NN EDA (Initial):



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	0	0	0	2	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	2	0	0	0	0	0	0
4	0	0	0	0	2	1	0	3	2	0	0	0	0	0
5	0	0	0	2	3	0	0	7	8	0	0	0	0	0
6	0	0	0	1	0	0	0	12	2	0	0	0	0	0
7	0	0	0	0	0	0	0	11	7	0	0	0	0	0
8	0	0	0	0	1	0	0	10	2	0	0	0	0	0
9	0	0	0	0	0	0	0	8	7	0	0	0	0	0
10	0	0	0	1	0	0	0	7	7	0	0	0	0	0
11	0	0	0	0	0	0	0	3	5	0	0	0	0	0
12	0	0	0	0	0	0	0	1	6	0	0	0	0	0
13	0	0	0	0	0	0	0	1	1	0	0	0	0	0

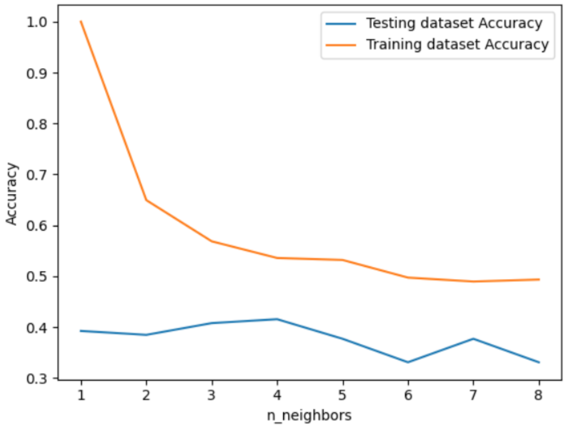
```
Mean Square Error: 8.285799056613284
R2 Score: 0.18701119909449848
Optimized Mean Squared Error: 8.222328244739105
Optimized R2 Score: 0.19323884942556735
```

Figure 1: Initial NN EDA Before Parameter Selection

This was the first iteration of the neural network, which we trained using attributes that we believed to be linearly correlated. Variables with the highest correlation to our target variable were selected, which intuitively made sense. The three attributes selected were 'failures', 'age', and 'G3', with 'failures' and 'age' as input attributes and 'G3' as the output variable. Subsequently, we developed the MLP regression neural network. After testing many learning rates, I found that the default learning rate of 0.01 yielded the best outcomes. A 1000 iterations were over double the amount used in the model examples provided to us in class. The final result was a neural network with the following scores (refer to MSE and R2 scores in Figure 1).

The high MSE and low R2 suggest that the neural network is not performing well. This could be because the model is underfitting and not capturing the data trends appropriately. It could also indicate that there is not a strong correlation between the variables (which is true since these attributes were linearly correlated in the 0.40 range). In one final attempt to improve scores, we ran grid search cross-validation over our neural network to tune our hyperparameters, although we received almost identical results (refer to Optimized MSE and R2 scores in Figure 1). Ultimately, as determined by our scores and confusion matrix displaying the extent of false predictions, we needed to completely reevaluate our approach to feature selection.

B. K Nearest Neighbors (Initial):



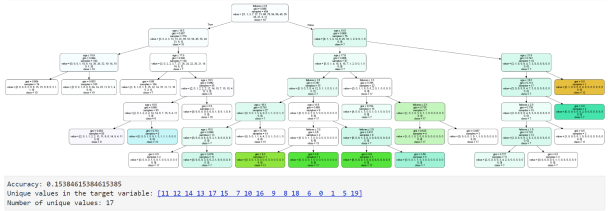
```
Accuracy: 37.69%
MSE: 2.8461538461538463
R2: 0.7081381519866022
```

Figure 2: Initial K Nearest Neighbors Before Parameter Selection

When running the K-near neighbors algorithm, we compute the test and training accuracies at each iteration of n_neighbors. From there, we pick the iteration of n_neighbors with the highest accuracies. Ideally, the training and test accuracies should converge at the n_neighbors we chose to verify a good fit for the model. However, as displayed in Figure 2, the results show a lack of convergence between the accuracies, which was the first indicator of poor performance.

After defining our visual model, we deduced that 7 n_neighbors seemed to be the closest convergence for both test and training. With that, we assigned 7 to our n_neighbors hyperparameter and made the KNN classifier to determine our final accuracy, MSE, and R2 scores. Even after tinkering with the said hyperparameter, our model attains both a high R2 score as well as a high MSE, which also indicates poor performance of the model. Specifically, it indicates the potential that our model is overly complex, resulting in overfitting that impacts our overall accuracy in our predictor. Thus, we concluded from our results that this model was not fit to be a predictor. However, from this model, we were able to conclude that complexity was an issue of our model, which eventually guided us to PCA to reduce the dimensionality of our dataset.

C. *Decision Tree:*



```
Accuracy: 0.15384615384615385
Unique values in the target variable: [11 12 14 13 17 15  7 10 16  9  8 18  6  0  1  5 19]
Number of unique values: 17
```

Figure 3: Decision Tree Before Parameter Selection

A Decision Tree was a third approach taken in an attempt to potentially classify our data and determine if it would yield solid results compared to the previous methods that took a regression approach. The data split that yielded the best results was a 70:30 split between training and testing respectfully. Similar to our EDA NN model, the three attributes selected were 'failures', 'age', and 'G3', with 'failures' and 'age' as input attributes and 'G3' as the output variable. As seen by the results of Figure 3, the accuracy of the classifier is particularly poor, even in comparison to the other models. After pre-pruning by establishing a max-depth of 3 and making the defining criterion entropy in an attempt to optimize the model, the highest performance attained through the model was an accuracy of 0.1692. Ultimately, not much knowledge was gained from this model, but we were able to conclude that classification would not be a good approach to our dataset.

D. *Principal Component Analysis:*

```
Highest correlations:
activities_yes  famsup_no         0.497973
famsup_no       activities_yes    0.497973
G3              activities_yes    0.419474
activities_yes  G3                0.419474
higher_no       activities_yes    0.368975
dtype: float64

Lowest correlations:
romantic_yes  paid_yes       −0.298532
              paid_no        −0.356810
paid_no       romantic_yes   −0.356810
              famsup_yes     −0.404820
famsup_yes    paid_no        −0.404820
dtype: float64
```

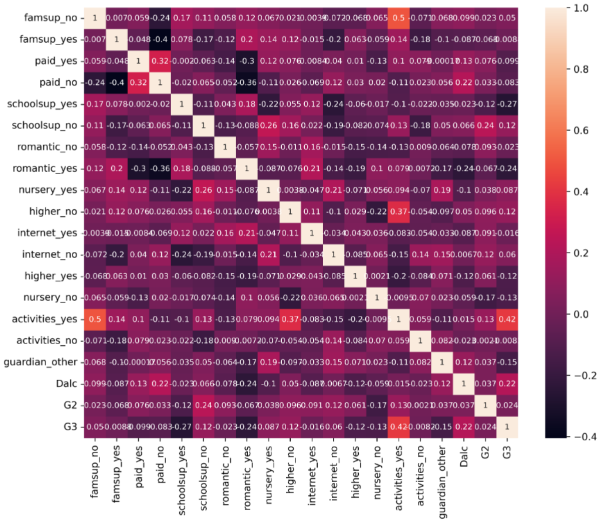


Figure 4: Principal Component Analysis Heatmap

From our initial experimental results and outside consultation, we deduced the most optimal way to improve results in our datasets was to run our dataset through a PCA. Specifically, we determined our data was far too complex to train an effective predictor model, meaning the best approach was to reduce the dimensionality of our dataset through Principal Component Analysis. PCA reduces the number of dimensions (features) in the dataset, leaving only the most significant data. This is especially helpful in simplifying our dataset, which had a large number of potentially correlated features. Referring to Figure 4, we see that the results of our feature selection created a far larger amount of more significantly correlated attributes compared to those of Figure 5, as well as reducing the number of features down to 20.

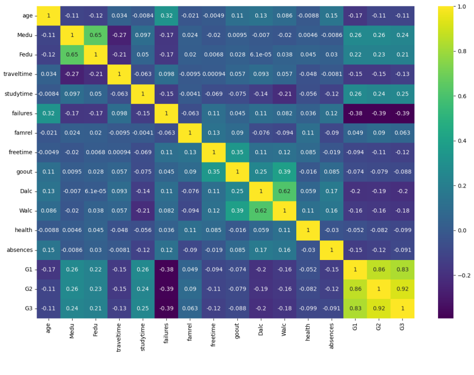
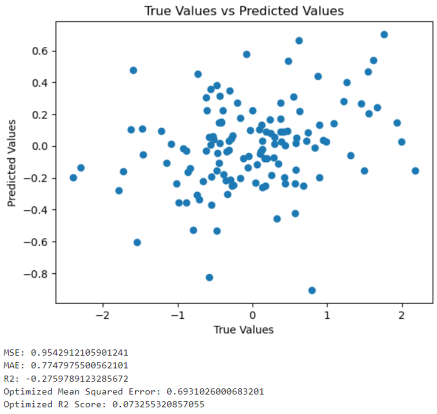


Figure 5: Original Dataset Heatmap

E. EDA NN (Post PCA):



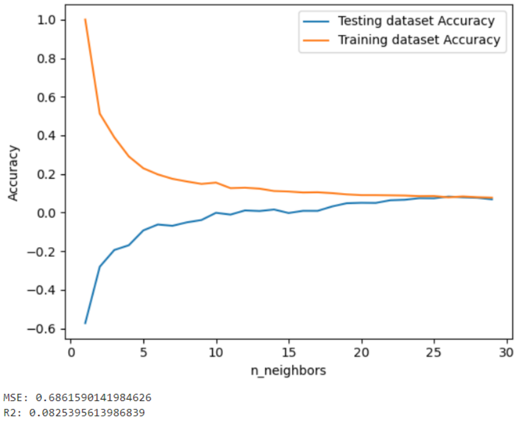
```
MSE: 0.9542912105901241
MAE: 0.774797550056 2101
R2: -0.275978912328 5672
Optimized Mean Squared Error: 0.6931026000683201
Optimized R2 Score: 0.073255320857055
```

Figure 6: Results of EDA NN by Comparing True to Predicted Values

On the retraining of our NN, we will train our NN based on these four attributes: Input: activities, famsup, higher; Output: G3. Note that the PCA includes continuous data, which is why we use an MLPRegressor. Ultimately, our approach to the retrial of our NN remains the same, the only thing that changed was how we modified our data. Even though we trained the NN with the results of our PCA, the model is still not very good. We manually selected the solver as ’sgd’ and the activation function as ’relu’, which both tend to default for an MLPRegressor. While our MSE is much improved, our R2 value is negative. This can be a sign of overfitting, as the model’s predictions are poor and could be overly complex.

Parameter tuning, which runs many iterations of the MLPregressor with different parameters, helps a great deal in this regard, as we can further minimize our MSE. However, a point of concern is that our R2 value is still quite low. This is why we ultimately did not decide to go with the NN as our final model.

F. K Nearest Neighbors (Post PCA):



```
MSE: 0.6861590141984626
R2: 0.0825395613986839
```

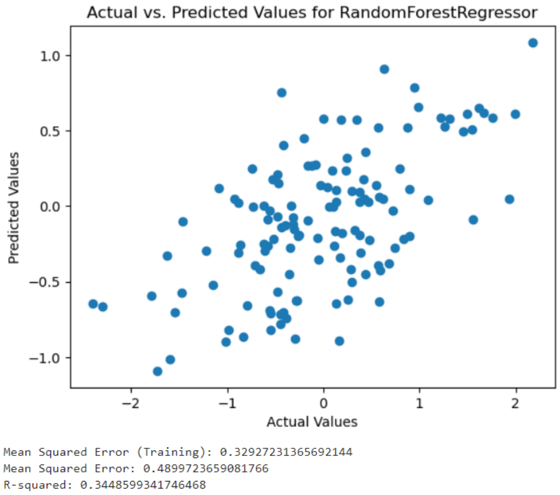
Figure 7: Initial K Nearest Neighbors Before Parameter Selection

When running the K-near neighbors algorithm, we compute the test and training accuracies at each iteration of n_neighbors. From there, we pick the iteration of n_neighbors with the highest accuracies. Ideally, the training and test accuracies should converge at the n_neighbors we chose to verify a good fit for the model. With the original unaltered data set, when running the model, we would receive an MSE of about 7.8, indicating a fairly inaccurate model.

After performing PCA on the data and reducing it to 20 attributes, we trained the model again. Visually comparing Figure 2 with Figure 7, we see that Figure 7 successfully converges unlike Figure 2, implying better performance. We see this model converge completely at 26, making that the most optimal value for our n_neighbors hyperparameter. Ultimately, this model received an MSE of about 0.81, indicating a large decrease in the errors present in the original model.

From there, we chose the attributes with the highest correlations based on a heat map: famsup_yes/no, activities_yes/no, and higher_yes/no, with G3 as the target variable. The MSE improved to about 0.68, which, while is still much improved, and shows a fairly accurate model, it did not perform quite as well as the Random Forest model, which is why we chose it over the K-near neighbors model.

G. Random Forest (Post PCA, Our Selected Predictor Model)



```
Mean Squared Error (Training): 0.32927231365692144
Mean Squared Error: 0.4899723659081766
R-squared: 0.3448599341746468
```

Figure 8: Random Forest Results

As previously mentioned in our methodology, the selection of Random Forest is based on its capacity to effectively handle non-linear associations within the data and deliver resilient predictions. Its use of multiple trees to make a regression allows for high accuracy, a lower chance of overfitting data, and a better handling of non-linear data. This makes the model on paper the most optimal choice for our principal components, which through PCA are meant to be decorrelated.

In regards to its performance, it ultimately performed the best of our three models. As seen in Figure 8, it, by comparison, has the best MSE and R2 score of the three models when compared to Figures 6 and 7. Initially, we were skeptical about the effectiveness of hyperparameter tuning for our model. However, upon reevaluating our approach, we decided to give it another try using Grid Search Cross Validation.

We established a hyperparameter grid, which included varying n estimators (50, 100, 200), max depth settings (none, 5, 10, 20), a range of minimum samples split (2, 5, 10), and different minimum samples leaf (1, 2, 4). We were pleased to find that hyperparameter tuning further improved our evaluation metrics of MSE and R2. The best hyperparameters emerged as the following: 'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200.

This optimal combination led to an 8.33% reduction in the Mean Squared Error (MSE) and an 8.76% increase in the R2 score, culminating in a final Testing MSE of 0.4926 and an R2 score of 0.4018. This performance improvement was substantial enough that we decided to adopt this as our final model for use in our web application, as these improved metrics suggest that our model is generalizing well, as indicated by the reduction in testing MSE and the increase in the R2 score, which would not be the case if overfitting had occurred.

VI. Conclusion and Discussion

This study focused on creating a machine-learning model to predict student grades. The model used extensive datasets from the student-mat.csv (Math course) and student-por.csv (Portuguese language course). At first, we looked into three different models - k-nearest neighbor, neural network, and decision tree - using all the features that were available. However, after noticing the significant Mean Squared Errors (MSEs), we decided to reevaluate our approach and opted to use Principal Component Analysis (PCA) for feature selection. Afterward, we created models using a smaller set of features, including k-nearest neighbors and neural networks and chose a random forest instead of a decision tree. After careful analysis, it was found that the random forest was the best option, showing the lowest MSE. After some fine-tuning, the results of our research paper showed a remarkable accuracy with an MSE of 0.490. This exploration highlights the significance of selecting relevant features and choosing appropriate models, resulting in a reliable predictive tool that shows potential for improving our comprehension of academic performance and enabling more informed educational interventions.

VII. Timeline (7 Weeks, Monday 11.16.23 – Friday 12.1.23)

- **Week 1 (10/16 – 10/22):**
  - *Deliverable:* Finalize road map.
  - *Deliverable:* Conduct a background literature review.
- **Week 2 (10/23 – 10/29):**
  - Conduct exploratory data analysis (EDA).
  - Visualize data distributions, correlations, and outliers.
  - Clean and format the data as needed.
- **Week 3 (10/30 – 11/5):**
  - Begin development of 3 regression/clustering model.
  - *Deliverable:* Complete the dataset description and exploratory data.
- **Week 4 (11/6 – 11/12):**
  - Complete the first iteration of the 3 ML models.
  - *Deliverable:* Complete the proposed methodology.
- **Week 5 (11/13 – 11/19):**
  - Optimize model performance.
  - Assess performance using various models.
- **Week 6 (11/20 – 11/26):**
  - Measure the models' final performance.
  - *Deliverable:* Complete experimental results, conclusions, and discussion.

- **Week 7 (11/27 – 12/1):**
  - *Deliverable:* Complete front-end web development.
  - Practice presentation and demo.
  - *Deliverable:* 12/1 Final submission.

Dataset:

UCI Machine Learning. (2016, November). Student Alcohol Consumption, Version 1. Retrieved December 4, 2023 from https://www.kaggle.com/datasets/uciml/student-alcohol-consumption

Literature Review:

[1] M. Nachouki, E. A. Mohamed, R. Mehdi, and M. A. Naaj, “Student course grade prediction using the random forest algorithm: Analysis of predictors’ importance,” *Trends in Neuroscience and Education*, vol. 33, p. 100214, 2023. [Online]. Available: https://doi.org/10.1016/j.tine.2023.100214

[2] S. A. B. Dianah *et al.*, “A predictive analytics model for students grade prediction by supervised machine learning,” in *IOP Conference Series: Materials Science and Engineering*, vol. 1051, 2021, p. 012005.