Implementing Random Forest Regressors For Predicting Educational Outcomes and Measuring Feature Importance

Leah Williams LCW4@WILLIAMS.EDU
Computer Science

1. Introduction

Not all students within a school or classroom respond to the same methods in an academic setting. Students may require an individualized approach to learning so it is hard to pinpoint what specific factors are best to look at. Existing methods of attempting to increase students' successes may place an over emphasis on the students' home life or academics depending on what the student or school official finds valuable. My goal for the project is to implement a machine learning model to identify the factors that lead to lower student success and in what circumstances it would need to be changed.

For this project student success is measured by students grades on the final exam. Different students may have different expectations for themselves and so looking at grades provides a continuous scale for success to be measured upon. Using grades puts emphasis on the individual's internal scale of success and a difference in someone's grades is easily quantifiable. In this project, I am analyzing how features influence an outcome variable with a continuous value, grades, rather than a binary classification because it provides a more precise measure of impact. Smaller changes are also better represented. Linear regression, decision trees and random forests will be used as models to predict students' grades. I predict that the random forests will perform them best because of the method of combining multiple different predictions to calculate a final prediction. I predict that a combination of factors outside the students control like their distance from the school and factors that represent their engagement in school like attendance have the greatest impact on their grades.

2. Preliminaries

Linear regression finds the best linear function that maps the vectors of examples X to an outcome variable Y. The matrix of examples X are multiplied by a weight or parameter vector to get a prediction $\widehat{Y}$. That is, for a given example, we have the following equation that determines how we obtain a prediction $\widehat{Y}$ as a function of inputs $x_1, \ldots, x_d$,

$$\widehat{Y} = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_3 x_d.$$

The weight vector is produced by finding the loss between the actual Y value and the predicted Y value, minimizing that loss and adjusting the weights appropriately over a certain number of iterations. The loss function is minimized using gradient descent. The

values of the final weight vector are an indicator of how much a particular value influences the final prediction.

When there is no linear boundary that can be drawn that perfectly separates two categories a decision tree can be used to develop a more complex boundary.

A tree is a data structure that consists of a set of nodes and a set of edges. A singular edge connects two different notes to each other. All nodes in a tree are connected to each other through a distinct path. Trees are also acyclic meaning that there are no cycles, so there is no path that starts and ends at the same node. A path is a unique sequence of vertices and edges in a graph.

A regression decision tree is a supervised learning technique that produces continuous valued output by following the path from the root to a leaf node and taking the average of the values in the leaf node. On a decision tree the nodes are a subsection of the data. Data points are moved from the root of the tree to the leaves according to a set of rules based on the loss functions. For regression the loss function is mean squared error. The predictions of a given node is the average of the outcomes.

$$MSE([D]) = \frac{1}{len([D])} \sum_{i=1}^{len([D])} (y_i - \bar{y})^2$$

The data is split by taking the weighted average of the split data set. The data is split along a feature that minimizes the mean squared error so that the child nodes will be in the most homogenous group possible with similar outcome variables. Whatever feature is chosen is presumed to best split the data between different outcomes and be a good indication of what features have the largest impact on the outcome.

Individual trees do not make the best models because they have the tendency to be over or under fit depending on the depth of the tree. A random forest is an ensemble of trees trained on the same dataset in an attempt to reduce the error from just one tree. All trees in a random forest have an unlimited depth. Each tree is semi-independent of each other and they all have a better than chance probability of being correct. When the predictions of those models are combined they have a better chance of being correct than an individual tree by itself. The prediction for a random regress forest is the mean of the outputs of the leaf nodes.

Bootstrap tree or bagging is a method to reduce bias when developing ensambles. Rather than having all the ensembles trained on the same data set, bootstrapping is used to generate new datasets from the original dataset. If a dataset has n number of examples, a new dataset is created by sampling those n examples uniformly with replacement. This method happens for each tree in the forest. The datasets are not fully independent but they are useful for reducing the overall error.

The scikit-learn implementation of linear regression and decision tree regressors and random forests are being used to implement these models.

Shapley values are being used to analyse the impact of the features on the model's prediction. Shapley values are the average marginal contribution of a feature across all possible conditions (Molnar (2025)). To simulate the marginal contribution of one feature,

all other features would be held constant and the difference in the prediction between including that feature and not including that feature is the contribution. Across multiple features all possible combinations of the features are found to identify the contribution of each individual feature.

The scikit-learn implementation of linear regression and decision tree regressor are being used to implement these models.

3. Data

The dataset was taken from a synthetic dataset on Kaggle, with the title "Student Performance Factors". The data has columns that record different aspects of a students academic life like the amount of hours of sleep they get and previous scores. There are also personal features like if they have a learning disability or their parent's highest education level.

The original dataset has 20 features and 6,607 examples. Some of the examples had missing data so the final dataset that was used consists of 3,825 examples with the same 20 features. The data did not have to be scaled because it was not necessary for the models being used, random forests, decision trees and linear regression. However, the scikit-learn model being implemented does not support categorical variables so all categorical values were converted to numeric values. Both the ordinal and nominal variables are converted into discrete variable numeric values.

A 70-15-15 split is being used for training, validation and testing. All of the features are being retained for the analysis, including the few demographic features there are. There are no student id numbers or names to omit. The demographic features like gender are kept because they may give insight to the culture around specific identities despite the fact that there are no interventions that can change one's identity.

4. Training And Validation Of Models

Rather than using accuracy to test the models I am using the mean squared error. Mean squared error measures the distance between the predicted value and the actual value and is always a positive value. The predictions that the models are making have up to eight decimal points so any small difference in the value will result in different accuracies. One drawback for the mean squared error is that there is no range that the error has to be between so there is no sense for the difference between errors. The mean squared error will be more harsh on larger differences between the prediction and true values than smaller ones. A better model is one that has a smaller mean square error.

I am using a linear regression model as my base line. If the random forests have a smaller mean squared error than the linear regression model that would determine the success of my machine learning model. The linear regression model has a final test error of 5.09.

To find the suitable depth of the decision tree, I looked at the mean squared error of the tree across multiple depths. A tree that is too shallow will be under-fit to the data and a tree that is too deep is overfit and may perform poorly on unseen data. When comparing

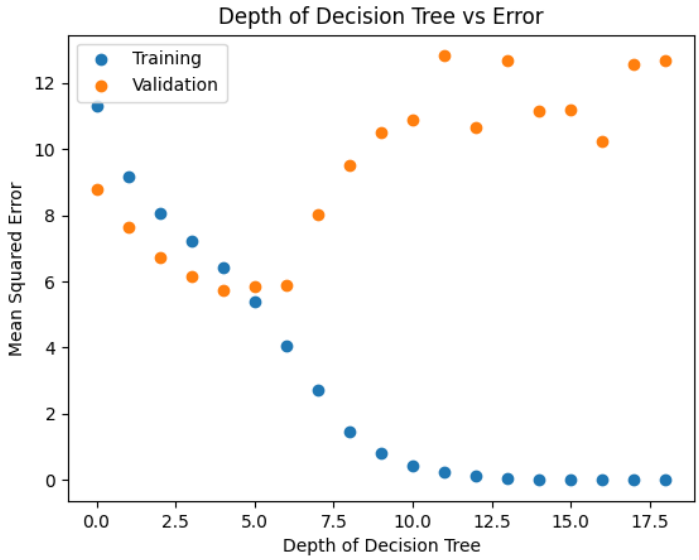

Figure 1: Decision Tree Comparing Performance on Training and Validation Sets

the training and validation errors, I found that a decision tree of depth 5 had the smallest validation error. The final validation error is 5.72

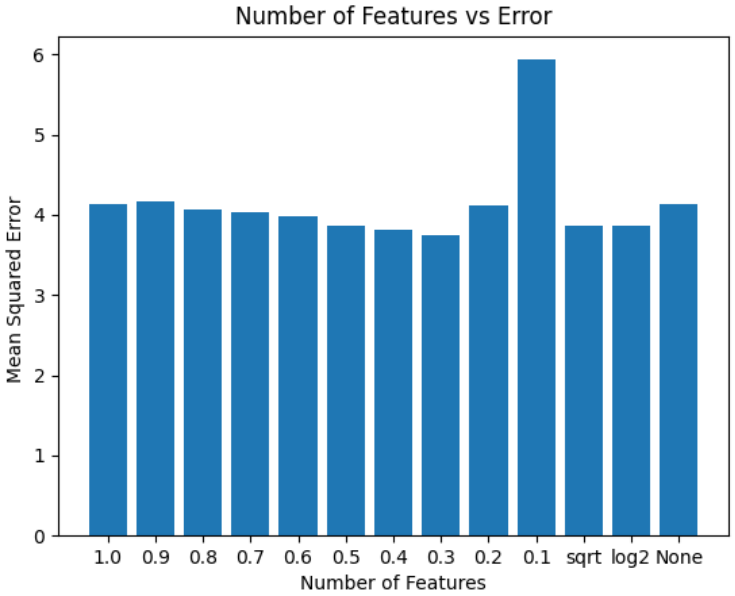

Figure 2: Mean Squared Error of Different Number of Features in Forest

For the random forest I chose there to be 500 trees in the forest because it is sufficiently large and within the computational power of the machine I am using. I also had to tune the amount of features that would be used to find the best split. Only considering 30% of the features had the smallest error. The final validation error is 3.87.

5. Results

The final test error for the linear model is 5.09. The test error for the decision tree is 10.90. The test error for the random forest is 6.52. The final test error of the random forest was greater than the error for the linear regression model. The random regression forest did not perform better than the baseline linear model. It did perform better than the single decision tree. The interpretability of the decision trees was not beneficial in providing accurate predictions. The forest was not able to capture the relationship between the features and the outcome variable as well as the linear regression model. This leads me to believe that for the dataset used there is a fairly linear relationship between the aspects of a students life and the grades they receive in school.

Figure 3: The Mean Squared Error of the Various Models Compared to Each other

6. Ablation Study

I performed an ablation study to see how the impact of certain features impacted the prediction of the model. The random forest found that the hours a student spends studying, their attendance in the class and the amount of access the student has to educational resources were the most important factors in their grade. The study looked at Shapley values for the different features, which analyzes the mode's predictions without those features along with different combinations of them to determine their impact in the final prediction. The Shapley values were only calculated for those three values because of the computational complexity of the process.

The model found that attendance was the most important feature for its prediction, followed by the amount of hours studied, then the students' access to resources. The Shapley values found for these features were in agreement with the determinations of the model.

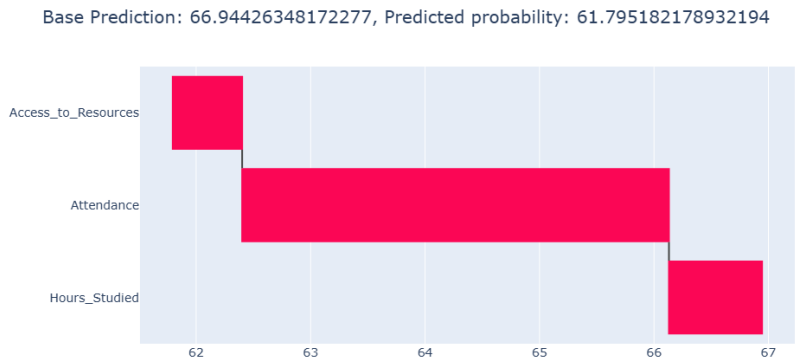

Figure 4: Waterfall graph of Shalepy Values

As seen in Figure 4 the relative values of the features were consistent with the determination of the importance of the feature by the model. A decrease in the access to resources, amount of time spent studying and attendance in class negatively impacts their grades in class.

7. Discussion and Conclusion

Despite the random forest not performing better than the linear regression model for this dataset it was still able to identify the features that most contributed to student grades. My prediction was incorrect that the random forest would out perform linear regression. Linear regression was able to capture the relationship between the features and student grades. For this dataset the relationship between the features of student life and grade are linear which the tree based models were not able to capture. In this case, linear regression was a faster and more accurate model than a decision tree or random forest.

The mean is highly sensitive to outliers and outliers can pull the mean toward their value. A single outlier could lead the mean to misrepresent the entire dataset. Since the prediction for the regression random forest and decision tree takes the average score in the leaf node as a prediction, if there were an outlier in the split of data then that could move the average farther than what it would be without that value. This process is happening for many layers for 500 trees, which can lead to the poor performance of the random forest compared to linear regression.

The decision tree pictures in Figure 5 shows the influence of a few features in the prediction of the tree as well as the sensitivity of the mean prediction. Many of the leaf nodes have similar values associated with them. Small differences in the mean still resulted

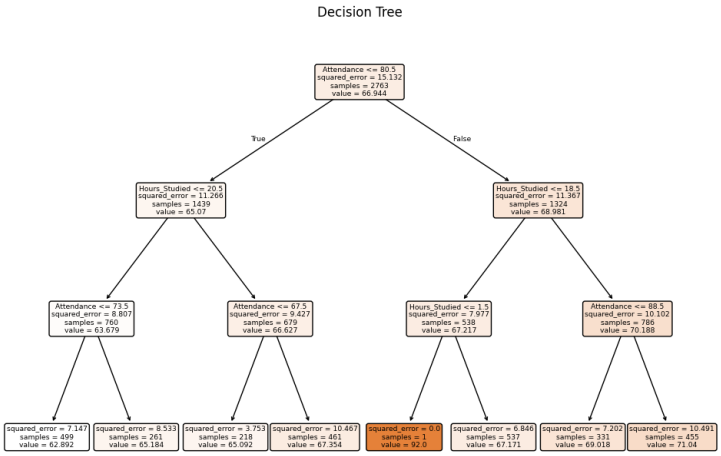

Figure 5: Decision Tree of Depth 3 to Visualize Decisions

in difference leaf nodes meaning that the sensitivity on the mean calculation impacts the prediction of the tree.

The random forest was useful in perceiving the relative importance of the characteristics in the outcome of the model. For this dataset, a student's attendance in class, amount of hours they spend studying, and access to educational resources were important factors on their exam score in the class. Being in class, studying and having appropriate access to outside help are crucial for student success. A combination of factors that are outside and within the students control contributed to their performance in class.

In future studies I would like to see these models' performance on a real world data set and how students' performance changed over time as a result of these factors. Implementing the models on real world data rather than synthetic data would inform how the models perform on data with more variability. If data were to be gathered on how students perform over time, it would be interesting to see if the same features remained important over multiple years.

References

Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub, 2025. URL https://christophm.github.io/interpretable-ml-book/.

ScikitLearn, MathplotLib and Plotly were used to create the linear models and graph the figures.