PREDICTING STUDENTS' ACADEMIC PERFORMANCE WITH DECISION TREE AND NEURAL NETWORK

by

JUNSHUAI FENG
B.S. Florida Gulf Coast University, 2017

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2019

ABSTRACT

Educational Data Mining (EDM) is a developing research field that involves many techniques to explore data relating to educational background. EDM can analyze and resolve educational data with computational methods to address educational questions. Similar to EDM, neural networks have been utilized in widespread and successful data mining applications. In this paper, synthetic datasets are employed since this paper aims to explore the methodologies such as decision tree classifiers and neural networks to predict student performance in the context of EDM. Firstly, it introduces EDM and some relative works that have been accomplished previously in this field along with their datasets and computational results. Then, it demonstrates how the synthetic student dataset is generated, analyzes some input attributes from the dataset such as gender and high school GPA, and delivers with some visualization results to determine which classification methods approaches are the most efficient. After testing the data with decision tree classifiers and neural networks methodologies, it concludes the effectiveness of both approaches in terms of the model evaluation performance as well as discussing some of the most promising future work of this research.

ACKNOWLEDGMENTS

I would like to express the deepest gratitude to my advisor Dr. Sumit Kumar Jha for his inestimable guidance, encouragement, and dedication throughout the course of this thesis's development. I would also like to thank Dr. Wei Zhang and Dr. Shaojie Zhang, who have willingly shared their precious time providing valuable comments and supports.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Prediction of students' academic performance has been one of the most popular Educational Data Mining (EDM) applications, and many different techniques have been implemented in this field since Data Mining can discover the hidden patterns in large educational databases [5]. Data mining has received a significant attention because of its meaningful impact in decision making and it has become an essential utility for varieties of companies. Besides, it has also been leading Artificial Intelligence, Machine Learning, Statistics, and Complex Computation into a new era [5].

By exploring the varieties types of data attributes that come from educational settings, EDM can help us understand students and the settings in which they learn. Predicting students' academic performance provides educational institutions opportunities to improve and maintain students' accomplishments during their presence in the institutions [14]. The main focus of EDM is to apply various methods to discover and extract hidden patterns or large stored database. Once these patterns are found, they can be further utilized to make effective decisions for the development of the educational institutions [5].

The main objective of this paper is to demonstrate how to apply statistical and machine learning techniques to predicting students' academic success. The dataset contains several aspects of the students' backgrounds such as gender and high-school grade point average (GPA). Most data attributes are generated based on the real statistical records gathered from The National Center for Education Statistics (NCES) [33], which is a principle federal organization that analyzes the insights and compares the educational data throughout the colleges and universities in the United States.

Since the dataset contains both continuous/numerical values and categorical/discrete values, classification technique can be utilized to categorize individual data sample based on the attributes

from a labeled training set. There are many classification algorithms in machine learning fields to implement EDM analysis such as Decision Trees, K Nearest Neighbor (kNN), Naive Bayes, Support Vector Machine (SVM) and Neural Networks [5]. This paper mainly focuses on decision tree classifiers and neural networks approaches to predict student academic performance and compares the effectiveness of both approaches in terms of modeling the evaluation performance.

CHAPTER 2: BACKGROUND

This chapter recapitulates some of the previous relative works on how Data Mining can be applied to educational field to enhance our understanding of learning process to focus on identifying, extracting and evaluating individual features related to the students' learning process [17].

In 2013, Kabakchieva conducted an research on predicting student performance with several Data Mining methods for classification such as Nave Bayes, Decision Tree, and k Nearest Neighbor (kNN) [24]. Kabakchieva utilized WEKA software to analyze the students' performance from University of National and World Economy (UNWE). WEKA is a popular software application that contains various of data analysis tools, such as data visualization on classification and clustering. The dataset used contains students' historical courses' grades, genders and ages, and the output of the students' final performance was categorized into five different levels, which are excellent, very good, good, average, and bad. The results revealed the significant factors influencing most the classification process are the students' University Admission Score and Number of Failures at the first-year university exams.

In 2008, Oladokun [36] and his team developed an Artificial Neural Network (ANN) model for predicting the likelihood of a candidate being considered for admission into the university from the National University Admission Examination System. In his experiment, 60% of the total sample data was used for network training, and 10% of the validation set is used to determine if the network is converging accurately for adequate generalization ability by checking the degree of learning network. Some input attributes include student' gender, age, parents' educational status, type and location of the secondary school. The output variable is based on the current grading system used by the university and the authors categorized the output into three different groups: Good, Average, and Poor as the performance level. In the testing phase, they optimized their model

by minimizing the Mean Square Error (MSE) of the computed results with the real student records and the model conducted an accuracy of 74%.

CHAPTER 3: DATA EXPLORATION

In this section, several data manipulation techniques are introduced to generate the synthetic educational dataset as well as the data visualization. The objective of the analysis and visualization of data is to highlight useful information and support decision making [40]. Due to the complexity of the student information, we limit the dataset size to 800 students' records for this experiment. To predict students' academic success, we first need to determine the potential factors that could cause impacts on the students' grades and behaviors. For instance, gender, first generation status, high-school GPA, and social activity status. Based on the statistical records from NCES, input features can be simulated with the real-world institution records.

Data Generation

In machine learning field, predicting students' academic performance is considered as supervised learning. Supervised learning summarizes the type of data mining when there are both input variables (x) and an output variable (Y), where (x) can be any student background aspect and output (y) represents the graduation performance of the student. The goal of this supervised learning is to find the best mapping function that can predict the output with the data we have [14].

The first step of the supervised learning is to organize the data into the way that can be easily read and processed such as aligning the output labels with all the input features in the same table. The implementation for data generation is accomplished in R. R is a programming language and environment for statistical computing and graphics including linear and nonlinear modelling, classical statistical tests, classification, clustering, and others [45]. As shown in Listing 3.1, we first generate all the feature inputs with specific ratios and then determine the final output label based

on those variables.

The first half section in Listing 3.1 demonstrates how each feature input is generated based on NCES statistics from the most college in the United States. The feature inputs include students' gender, first generation status, physical activity status, high school GPA, and whether they are transfer students from other universities. All the inputs are filled in a table named `student`, and `graduateBool` is set as the output labels.

Listing 3.1: Synthetic data implementation

```
set.seed(2)
dataSize = 800
Gender <- sample(c('F','M'), dataSize, replace=T, prob=c(11,9))
First.Generation.Flag <- sample(c(0,1), dataSize, replace=T, prob=c(4,1))
Transfer <- sample(c(0,1), dataSize, replace=T, prob=c(86,14))
Gym <- sample(c(0,1), dataSize, replace=T, prob=c(2,3))
library(truncnorm) #help setting GPA upper bound
HS.GPA <- rtruncnorm(dataSize, a=0, b=5, mean = 3.8, sd = 0.4)

weightSum <- (First.Generation.Flag*0.6 + Transfer*0.6 + Gym*0.7 + HS.GPA*0.8)
students <- cbind.data.frame(Gender, First.Generation.Flag, Transfer, Gym, HS.
    GPA)

total.graduate.rate <- 0.69 #based on real stats
bound <- quantile(weightSum, 1-total.graduate.rate)
students$graduateBool <- with(students, ifelse(weightSum>bound, 'Y', 'N'))
```

The ratio distribution for the feature inputs are simulated from real statistical records from NCES. For instance, NCES reported that in fall 2017, the majority of college and university are expected to have more female than male students, and the statistical records for both genders are about 11.5 millions and 8.9 millions respectively [33]. To match the same gender ratio from the majority

institutions, the students' gender proportion for this research follows a ratio of 11:9 for female and male respectively. As illustrated in the pie chart 3.1, the female percentage is marginally greater than the male proportion. The approximate ratio between the numbers of female and male students is about 11:9 respectively in the synthetic dataset.

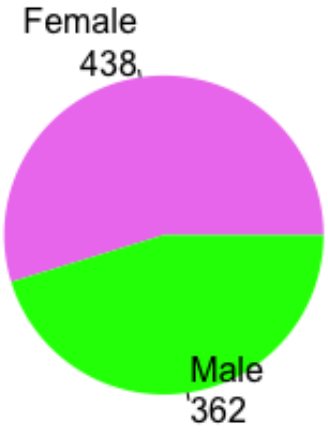

Figure 3.1: Gender Comparison

The output label `graduateBool` is calculated based on the feature records from each student and placed in the last column of the `student` table. According to The Integrated Postsecondary Education Data System (IPEDS) from NCES, the graduation rate for the undergraduate students from the University of Central Florida (UCF) is 69% [1]. For each student record, we yield different weight for each feature as multivariate linear regression model, split the distribution with the top 69% percentile records, and label the upper and lower partitions as `Y` and `N` respectively, which represents whether the students can obtain their bachelor's degrees.

Data Cross Validation

In supervised training, dataset can be divided into three categories: training set, validation set and testing set, which is also known as cross validation [3]. The training set allows the system to observe the type of relationships between various input data features and the outputs, which in our case is the graduating rate. After the training process, it generates and conducts a model that integrates the features relationship to the output labels [34]. This model can be a simple logistic equation that represents the dataset's correlations. It can also be a complex equation filled with many hyper-parameters such as weight for each feature, learning rate for each training iteration in neural network.

In order to estimate how well the trained model fits in our data, we need validation dataset and test dataset. The validation set is used to prevent overfitting and estimate prediction error for model selection, while the test set is to measure how well the final model can achieve [34]. However, in many algorithms, only one of these two datasets is adequate to give the model directions that can reduce the cost and error, and increase its accuracy [3].

Listing 3.2: Data cross validation

```
library(caret)
set.seed(2)
trainDB <- createDataPartition(students$graduateBool, p=0.7, list = FALSE)
testDB = -trainDB
training_DB = students[trainDB,]
testing_DB = students[testDB,]
```

Code description: With `seed(2)` function, we are able to randomize our dataset into arbitrary sequence and obtain the same generated values, which reduces the bias error during training and prevent overfitting.

The data processing phase was implemented under the condition that all the students' record had to be randomized first so the model can avoid gathering students with similar background. After organizing the data to an appropriate size, the dataset was split into training set and testing set with a ratio of 7:3 as shown in Listing 3.2. With the cross validation method, the model can be trained with `training_DB`, which has selected features such as students' genders and high school GPAs, and then use the `testing_DB` to compute the accuracy from the trained model.

CHAPTER 4: STATISTICAL METHODOLOGIES

Statistics is a mathematical science concerning the collection, analysis, interpretation or explanation, and presentation of data [12]. With this expressive analysis, we can provide such comprehensive data insights and summaries about learner's behavior from the educational data [49].

High School GPA & Graduation Rate

Norvilitis and Reid [35] conducted that students' high school success has a substantial positive correlation with their college success, as measured by their college GPAs, and this trend continues to be effective to the prediction of the successive academic success. Though given differences in grading standards across high schools, GPA may not provide a consistent measure of a student's ability in mathematics, reading, and other subjects. but GPA usually captures whether a student consistently attends class and completes her assignments on time [9]. In addition, Chingos argued that student's academic preparation in high school is one of the strongest predictors of college degree attainment. and those students with higher high school GPAs are much more likely to graduate in college [8].

Since high school GPA plays an essential role in college graduation rate, we assign the high school GPA feature a weight factor of 0.8 for the multivariable linear regression model as shown in Listing 3.1. Figure 4.1 illustrates the correlation between students' high school GPAs and their corresponding college graduation status for our synthetic dataset. We can see the median for college graduates have higher high school GPA than the non-graduate students, which simulates the statistics from Chingos [8].

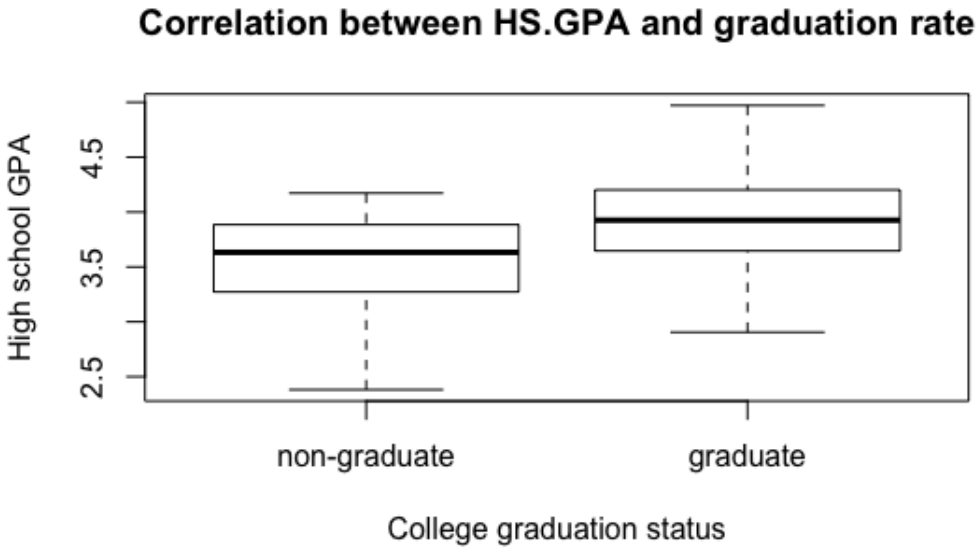

Figure 4.1: High-school GPA Distribution in Synthetic Dataset

CHAPTER 5: MACHINE LEARNING METHODOLOGIES

In this section, we explore two popular classification algorithms: decision tree classifiers and neural networks. Classification technique has been one of the common applied algorithms for data mining. It employs big data with pre-classified examples and then conducts a model that can predict and classify the large amount of the new data records [5]. This approach frequently employs several popular algorithms, such as Naive Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine and Logistic Regression. The data classification process involves both learning and classification phases. First, the algorithm analyzes the training data in the learning phase, then for the second phase, it applies the test data to measure the performance of the classification model generated from the first phase. The classification model from this classifier-training algorithm includes a set of parameters that best fit the pre-classified training examples. Thi model that encodes these parameters is called a classifier [31].

Decision Tree Classifiers

The decision tree is a suitable and practical method to build classification model since it is relatively fast, and can be easily converted to simple classification rules [2]. It is a supervised predictive modelling approach that uses a tree-like structure for its representation. The decision tree method mostly utilizes the information gain and the gini index to determine the attribute that is most useful [42]. The information gain is used when the data attributes are categorical, and for the gini index, attributes are assumed to be continuous.

The information gain measures the information contained by each attribute since each attribute has different impact on the output. In order to calculate the information gain, we need to compute

the entropy measure for each attribute. Gain calculates the expected reduction in entropy due to sorting on the attribute. For example, in Figure 5.1, the attribute that has the highest gain in this experiment is the high school GPA. This attribute is considered as the root node of the decision tree. The process is repeated for the remaining attributes to build the next level of the tree. The final tree structure consists of the possible scenarios and the outcomes that are represented by each individual tree branch [5].

Entropy measures the pureness of a table. For example, if a table contains only one class, it is considered as pure table as shown in equation 5.1, then the probability becomes 1 and log(1) = 0. In contrast, if all classes in the table all have equal probability, then the entropy measure becomes maximum, which is 1 [5]. Therefore, lower entropy value returns higher information gain, and the attribute with highest information gain is chosen as the splitting attribute for node N [47].

$$Entropy = -\sum_{i=1}^{C} p_i \, log_2 \, p_i \tag{5.1}$$

The gini index is a metric to measure how often a randomly chosen element would be incorrectly identified. Gini index can be summarized as equation 5.2. Similar to entropy measure, in a pure table, which consists of only one single class, the gini index is 0 since the probability for that class to occur is 1, and $1 - 1^2 = 0$. Furthermore, when the occurrence of all the labeled classes in the table are equivalent, the gini index reaches its maximum value [5].

$$\text{Gini Index} = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{5.2}$$

Decision Tree Result Visualization

Decision tree can be implemented and plotted with `rpart` package in R. Listing 5.1 demonstrates how to build the tree to predict the graduation rate from all the input attributes in `training_DB`, which contains 70% of the overall randomized synthetic data. We use Analysis of Variance (ANOVA) method to measure the correlations between the features and the output. ANOVA is a statistical method used to test differences between the means from the input features in order to determine whether there are any statistically significant differences between the means of features and the output.

Listing 5.1: Ploting Tree with Rpart

```
library("rpart")
library("rpart.plot")
rpartDB <- rpart(graduateBool~., data = training_DB, method = "class")
rpart.plot(rpartDB, extra=4, type = 3, tweak = 1.2, roundint=FALSE)
```

Code description:

- `rpartDB` computes the information gains from all attributes and sorts their importance in tree structure.
- `rpart.plot` library helps illustrate the tree structure visually.

Figure 5.1 is generated with the `rpart.plot` function from Listing 5.1. It projects the trained model with corresponding decision making at each node, and the overall likelihood of student's performance is displayed at the bottom of each tree branch. The color gradient of bottom level nodes indicates the student who has the background pattern from that branch is likely to pass or fail from obtaining the college degree. For instance, at the top level, if a student's high school GPA is greater than 3.5, then the student will most likely to graduate college, and his or her graduation rate is 89%. Green color represents a scenario of success, and the stronger the color is, the more likely it is to fall into its labeled output. In contrast, blue color indicates the opposite situation.

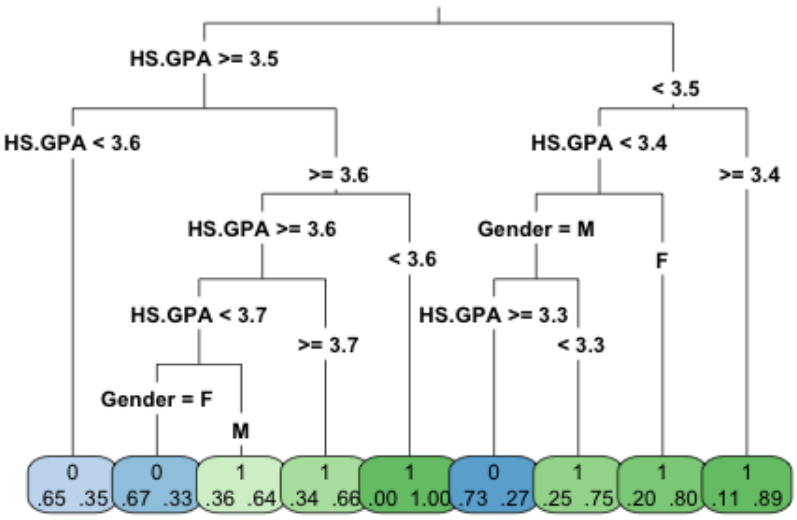

Figure 5.1: Decision Tree Generated with Rpart.plot Package

Artificial Neural Networks

Inspired by the biological neural networks that constitute our human brains, Artificial Neural Networks (ANN) systems mimic the logic behind it by establishing an array of interconnected nodes that exchange information among each other as illustrated in Figure 5.2 [14]. The input, hidden, and output layers derived from the structure of our neurons, which get signals from dendrites and output the signals with axons, in order to exchange information. Similar to how children can pick up skills from observing their parents, the neural networks can learn to perform varieties of tasks by considering and observing new examples, generally without being hard coded programs and rule sets [20]. Unlike Decision Tree Classifier, Neural Network can optimize its model iteratively and summarize the data in a parallel way just like how we human can perform thinking, looking, and moving body parts at the same time.

Figure 5.2 illustrates a standard three-layer neural networks interconnected with group of nodes.

Each circular node in the figure denotes an artificial neuron and the arrows crossing among these nodes represent the signal connection that passes information from one neuron to another [20]. In general, the signal of the neuron connections, also known as 'edge', is a real number, and based on its value and some non-linear function, we can calculate and obtain the new information and pass it into the next network layer as output [4]. The function typically assigns a fine-tuned weight that adjusts those artificial neurons as learning proceeds. During the learning process, the neural network learns and improves itself by adjusting weights and other parameters in order to be capable of predicting the correct class labels of the new input features [5]. The weight increases or decreases the strength of the signal at a connection. Besides, different layers in ANN may also perform different kinds of transformations on their inputs [4].

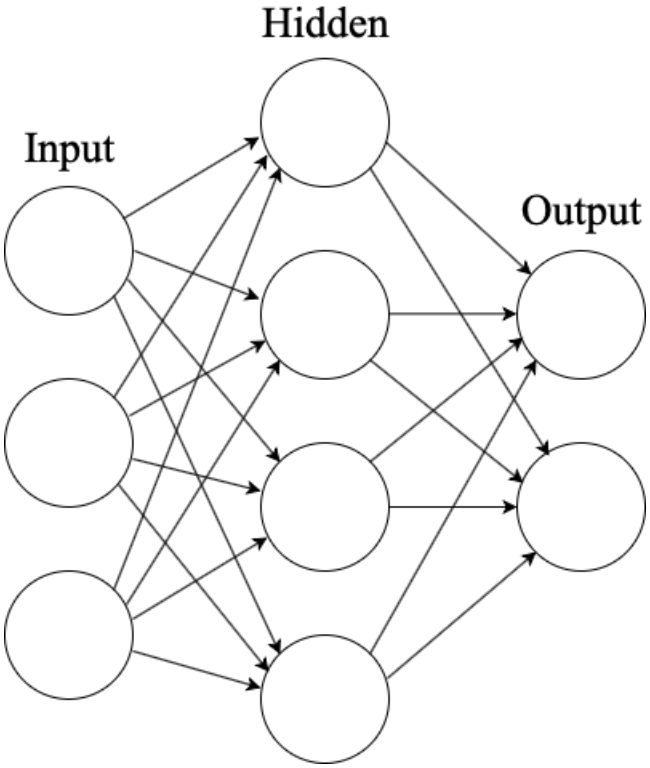

Figure 5.2: Standard Neural Networks Structure

With this sophisticated structure, ANN is well known for solving complex application in numerous business applications, such as predicting stock trends, translating human speech, and playing chess again real human beings. The reason why it is so powerful is that is has the ability to detect all possible correlations between the input features. Furthermore, it could also perform a complete detection even when the dataset contains sophisticated nonlinear relationships between predictors variables [43]. All these unique properties make ANN particularly suitable for a domain like EDM with large noisy dataset.

Neural Networks Result Visualization

In this experiment, we train the neural networks model using `neuralnet` package in RStudio. In `neuralnet` package, the back-propagation algorithm and three versions of resilient backpropagation are implemented, and they provide a custom-choice of activation and error function [21]. We use the sum of squared errors (SSE) cost function metric to measure how well the model performs. SSE is the sum of the squares of residuals that measures the discrepancy between the real data and an estimation model. The weight for each attribute can be updated and optimized by reducing the SSE.

Figure 5.3 reflects the structure of the trained neural network implemented using $neuralnet$ package. On the left, it lists different features that are fit into training the model as input layer. The hidden layer in the middle is computed with $y(x_i) = \sum(w_i x_i + b_i)$, where each hidden node x_i gathers the information from all the input attributes with corresponding weights w_i. The numbers next to each arrows are the weight values. The blue node in the graph represents the bias b_i added into each process. These network parameters are adjusted based on the back-propagation [41]. Finally, the output node on the right is calculated with the activation function and it indicates the decision output of the student academic performance. The result for the overall SSE and the

numbers of computing steps during the training process are labeled at the bottom of the graph.

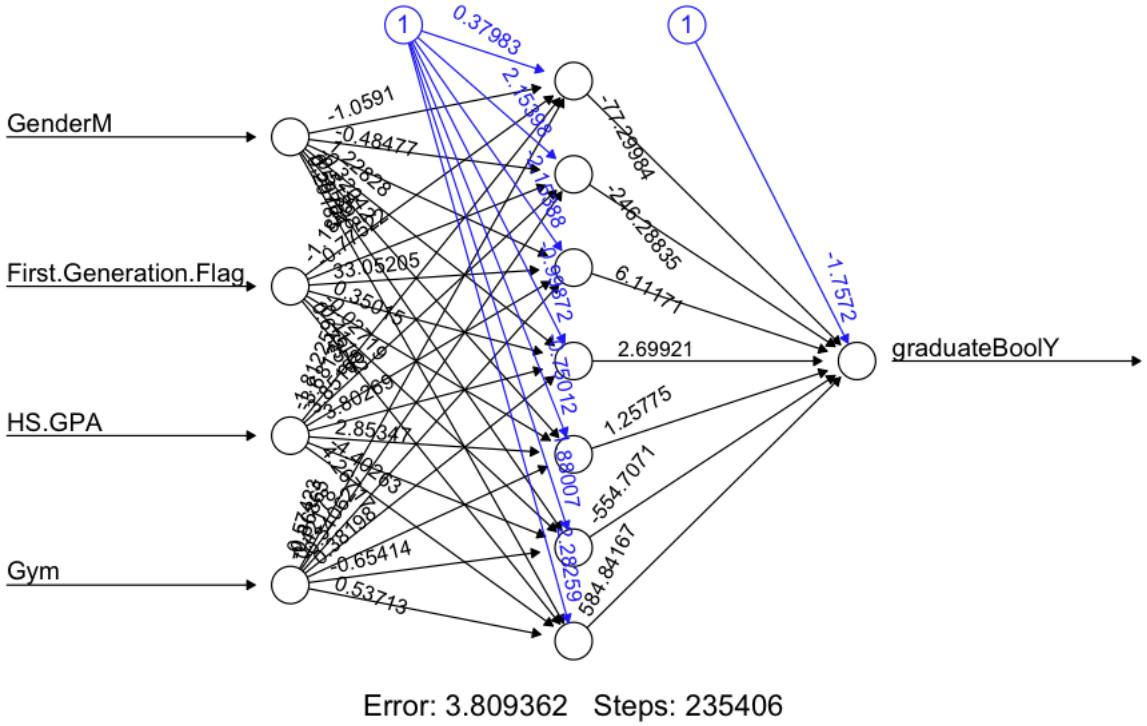

Figure 5.3: Neural Networks Output

CHAPTER 6: EVALUATION AND RESULT DISCUSSION

The testing phase in this experiment utilizes the confusion matrix in order to compute the accuracies of the models. As shown in Table 6.1, the confusion matrix is a simple $n \times n$ dimension table describing the performance of a classification model by comparing the predicted results with the actual results. In our case, $n = 2$ since our predicting outputs have two classes. With the confusion matrix, we can obtain the accuracy by adding the true positive and true negative values in the table and divide by the total data points as shown in the equation 6.1.

Table 6.1: Confusion Matrix

		Predicted class	
		Yes	**No**
Actual class	**Yes**	True Positive	False Negative
	No	False Positive	True Negative

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.1}$$

The confusion matrix result for the decision tree is projected on the left side of Table 6.2. With the equation 6.1, we can compute the accuracy of the model to be 92.8% as shown in Table 6.3. We can also improve the model accuracy by pruning the decision tree. The benefit of pruning a decision tree is that it removes the tree branches that contribute little to the overall classification process and reduces the tree size [15]. Besides, it simplifies the complexity of the final model classifier, reduce the likelihood of model overfitting, and improves the predictive accuracy [30].

Table 6.2: Confusion Matrix Results for Decision Tree (Left) & Neural Network (Right)

		Predicted class: Yes	Predicted class: No
Actual class	Yes	157	8
	No	6	68

		Predicted class: Yes	Predicted class: No
Actual class	Yes	159	2
	No	3	76

Table 6.3: Accuracies of Decision Tree Classifier Model

	Before Pruning	After Pruning
Accuracy	92.8%	94.1%

There are several principal methods for pruning decision trees, such as Error-Complexity Pruning, Critical Value Pruning, and Reduced-Error Pruning [30]. The method used in this experiment is Cost-Complexity Pruning, which is similar to Error-Complexity Pruning. It takes account of both the number of misclassification errors and the complexity (size) of the tree. Before pruning, each node in the leaves contains only one class, therefore there's no misclassification cost. After we prune the sub-trees that are less that certain threshold, the new terminal nodes might include classes from other branches, which is where misclassification error occurs. By adjusting the information gain threshold, we can obtain the best error cost of the sub-trees from training samples [29].

The complexity parameter (cp) in `rpart` package is the minimum improvement in the model needed at each node and it's based on the cost complexity of the model [46]. The `plotcp` function automatically illustrates the cross-validation cost results as the changes of the complexity parameter and the size of the tree. We can optimize the performance of the decision tree pruning result by choosing the cp value with the minimum error. After pruning the tree, our model accuracy

increases up to 94.1% as shown in Table 6.3.

Comparing to the decision tree performance, the accuracy for neural networks model is marginally higher, which is 97.6% on average as shown in Table 6.4. The table includes testing results of different parameters while keeping other training parameters the same, such as the back-propagation algorithm, SSE cost function, stepmax value and learning rate. From the results in Table 6.4, we can see the number of neural nodes and layers do not have much impacts on the model accuracy respecting to our synthetic dataset.

Table 6.4: Accuracies of Neural Network Model with Different Parameters

Number of nodes	Number of layers	Accuracy
4	1	97.9%
5	1	97.5%
6	1	97.5%
7	1	97.9%
4	2	97.5%
5	2	97.5%
6	2	97.9%
7	2	97.1%

The main reason why we can obtain such promising accuracies for both classification algorithms is because of our synthetic dataset. Unlike the real-world data, our dataset is much simpler and does not have much noises and outliers, which can significantly increase the prediction accuracy. Neural networks outperforms the decision tree classifiers in our experiment because it has noteworthy abilities to detect meaningful insights from large complex and imprecise data, extract their hidden patterns, and discover the data trends that cannot be easily perceived by either humans or other computing algorithms [5], though it is more suited for continuous valued inputs. The reason why decision tree can perform almost as well as neural network algorithm in this experiment is because it is capable of analyzing both numerical and categorical data inputs. It can also handle large amount of dataset and its classifier can be easily comprehended and interpreted the correlations

among the predictors variables, especially categorical data [43].

CHAPTER 7: CONCLUSION AND FUTURE WORK

This paper surveyed some of the most relevant work in this area, delivered graphical visualization for some input attributes, and developed decision tree and neural network algorithms with our synthetic dataset respecting to EDM. In this experiment, both decision tree classifiers and neural networks perform well interpreting the relationships between the variables and predicting student academic performance. We conclude that it is important to choose a classification model that is suitable for various types and complexity of the dataset.

For future experiments, it is expected to increase the complexity and more useful students' data. In this way, we could take fully advantage of neural network classifier's ability and measure how different those data attributes can affect the model's performance. It can also be applied with some realistic educational data, which would contain some outliers and noises. Additional machine learning algorithms can also have the potentials to compute an enhanced performance, such as Random Forest, Nave Bayes, and Support Vector Machines.

Consequently, the topic of predicting students' performance with data mining technique has inspired many researchers to continue their researches. It will benefit the educational institutions and industries to monitor and assist students' performance in a more efficient approach [43].

APPENDIX : R SOURCE CODE

```
# set running environment and read 'students' dataset if exists

setwd(dirname(rstudioapi::getSourceEditorContext()$path))
if (file.exists("students.csv")){
  students <- read.table("students.csv", sep = ",", fill=TRUE, header=T)
  attach(students)
  } else {
  cat('Please generate the synthetic code first.')
}

##################################################

#Generating synthetic student data
#No need to run this section if 'students' dataset exists
set.seed(2)
dataSize = 800
Gender <- sample(c('F','M'), dataSize, replace=T, prob=c(11,9))
First.Generation.Flag <- sample(c(0,1), dataSize, replace=T, prob=c(4,1))
Transfer <- sample(c(0,1), dataSize, replace=T, prob=c(86,14))
Gym <- sample(c(0,1), dataSize, replace=T, prob=c(2,3))
library(truncnorm) #help setting GPA upper bound
HS.GPA <- rtruncnorm(dataSize, a=0, b=5, mean = 3.8, sd = 0.4)

weightSum <- (First.Generation.Flag*0.6 + Transfer*0.6 + Gym*0.7 + HS.GPA*0.8)
students <- cbind.data.frame(Gender, First.Generation.Flag, Transfer, Gym, HS.
    GPA)

total.graduate.rate <- 0.69 #based on real stats
bound <- quantile(weightSum, 1-total.graduate.rate)
students$graduateBool <- with(students, ifelse(weightSum>bound, 'Y', 'N'))
```

```
# View(students) #uncomment to view the dataset

# Save 'students' dataframe into a local CSV file
write.csv(students, file = "students.csv", row.names=FALSE)

##################################################

# Gender visualization
genderProb <- table(students$Gender)
labs <- paste0(c('Female', 'Male'), "\n", genderProb, "\n")
pie(genderProb, labels = labs, border = F, col = c("violet", "green"))

# More feature comparisons
db <- data.frame(cbind(table(First.Generation.Flag), table(Transfer), table(
    Gym)))
names(db) <- c('First.Generation.Flag', 'Transfer', 'Gym')
barplot(as.matrix(db), main="More feature comparisons",
        ylab="Number of students", beside=TRUE, col=c('lightcyan','green'))
legend('topright', legend=c('No','Yes'),cex=0.7,horiz=T,
       fill=c('lightcyan','green'))

##################################################

# HS GPA vs graduation rate
GPA <- subset(students, select = c(HS.GPA, graduateBool))
boxplot(HS.GPA ~graduateBool, GPA, names=c('non-graduate', 'graduate'),
        ylab='High school GPA', xlab='College graduation status',
        main='Correlation between HS.GPA and graduation status')

##################################################

# Decision tree
```

```
#Cross validation
library(caret)
set.seed(2)
trainDB <- createDataPartition(students$graduateBool, p=0.7, list = FALSE)
testDB = -trainDB
training_DB = students[trainDB,]
testing_DB = students[testDB,]

library("rpart")
library("rpart.plot")
rpartDB <- rpart(graduateBool~., data = training_DB, method = "class")

#get accuracy
getAccuracy <- function(fit) {
  predictLabel <- predict(fit, testing_DB, type = 'class')
  matrixTable <- table(testing_DB$graduateBool, predictLabel)
  print('Confusion matrix for DT: ')
  print(matrixTable)
  accuracy_Test <- sum(diag(matrixTable)) / sum(matrixTable)
  cat('Accuracy for DT: ', accuracy_Test*100, "%")
}
rpart.plot(rpartDB, extra=4, type = 3, tweak = 1.2, roundint=FALSE)
getAccuracy(rpartDB)

#prune tree
plotcp(rpartDB) #find the best/lowest cp value
control <- rpart.control(cp = 0.011) #replace value with the lowest cp
prunedDB <- rpart(graduateBool~., data = training_DB, control = control,
    method = "class")
getAccuracy(prunedDB)
```

```
################################################

# NN - neuralNet
nnData <- students

# normalize HS GPA
normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }
nnData$HS.GPA <- normalize(nnData$HS.GPA)
nnData$graduateBool <- as.factor(nnData$graduateBool)



#Create dummy variables and remove the intercept
nnData <- data.frame(model.matrix(
  ~graduateBool+Gender+First.Generation.Flag+HS.GPA+Gym, data = nnData)[,-1])

#Cross validation
library(caret)
set.seed(2)
trainIndex <- createDataPartition(nnData$graduateBool, p=0.7, list = FALSE)
training_DB = nnData[trainIndex,]
testing_DB = nnData[-trainIndex,]

library(neuralnet)
col_list <- paste(c(colnames(training_DB[,-1])),collapse="+")
f <- formula(paste0("graduateBoolY~",col_list))
nn <- neuralnet(f, data = training_DB, hidden=7, err.fct = "sse",
                lifesign="minimum", linear.output=F, stepmax=1e6, rep=2)

plot(nn) #plot nn

NNoutput <- compute(nn, testing_DB[,-1])
```

```
pred <- ifelse(NNoutput$net.result > 0.5, 1, 0)
nnCM <- table(pred, data.matrix(testing_DB[,1])) #confusion matrix
print('Confusion matrix for NN: ')
print(nnCM)
NNaccuracy <- sum(diag(nnCM))/sum(nnCM) #accuracy
cat('Accuracy for NN: ', NNaccuracy*100, "%")
```

LIST OF REFERENCES

[1] IPEDS DATA FEEDBACK REPORT 2017. National center for education statistics. Retrieved from https://nces.ed.gov/ipeds/DataCenter/DfrFiles/IPEDSDFR2017_132903.pdf.

[2] Qasem A Al-Radaideh, Emad M Al-Shawakfa, and Mustafa I Al-Najjar. Mining student data using decision trees. In *International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan*, 2006.

[3] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statist. Surv.*, 4:40–79, 2010.

[4] Artificial neural network. Artificial neural network — Wikipedia, the free encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Artificial_neural_network [Online; accessed 23-October-2018].

[5] Brijesh Kumar Bhardwaj and Saurabh Pal. Data mining: A prediction for performance improvement using classification. *CoRR*, abs/1201.3418, 2012.

[6] Markus Brameier and Wolfgang Banzhaf. A comparison of linear genetic programming and neural networks in medical data mining. *IEEE Transactions on Evolutionary Computation*, 5(1):17–26, 2001.

[7] R Brause, T Langsdorf, and Michael Hepp. Neural data mining for credit card fraud detection. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 103–106. IEEE, 1999.

[8] Matthew M. Chingos. What matters most for college completion? academic preparation is a key of success - third way. Retrieved from https://www.thirdway.org/report/what-matters-most-for-college-completion-academic-preparation-is-a-key-of-success.

[9] Preston Cooper. What predicts college completion? high school gpa beats sat score, Jun 2018.

[10] Paulo Cortez. Data mining with neural networks and support vector machines using the r/rminer tool. In *Industrial Conference on Data Mining*, pages 572–583. Springer, 2010.

[11] Paulo Cortez and Alice Maria Gonçalves Silva. Using data mining to predict secondary school student performance. 2008.

[12] R. Purves D. Freedman and R. Pisani. Statistics, 4th ed. 2007.

[13] Bedasree Das. Us admission cycle: Know all about spring, summer and fall. 2015. Retrieved from `https://studyabroad.careers360.com/articles/us-admission-cycle-know-all-about-spring-summer-and-fall`.

[14] Ryan Shaun Joazeiro de Baker and Adriana M. J. B. de Carvalho. Labeling student behavior faster and more precisely with text replays. In *Educational Data Mining 2008, The 1st International Conference on Educational Data Mining, Montreal, Québec, Canada, June 20-21, 2008. Proceedings*, pages 38–47, 2008.

[15] Hepu Deng, Duoqian Miao, Fu Lee Wang, and Jingsheng Lei. *Emerging Research in Artificial Intelligence and ComputationaI Intelligence: International Conference, AICI 2011, Taiyuan, China, September 23-25, 2011. Proceedings*, volume 237. Springer, 2011.

[16] A. Dutt, M. A. Ismail, and T. Herawan. A systematic review on educational data mining. *IEEE Access*, 5:15991–16005, 2017.

[17] Alaa el Halees. Mining students data to analyze e-learning behavior: A case study. 2009.

[18] W. W. T. Fok, Y. S. He, H. H. A. Yeung, K. Y. Law, K. Cheung, Y. Ai, and P. Ho. Prediction model for students' future development by deep learning and tensorflow artificial intelligence engine. In *2018 4th International Conference on Information Management (ICIM)*, pages 103–106, May 2018.

[19] T. D. Gedeon and S. Turner. Explaining student grades predicted by a neural network. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 1, pages 609–612 vol.1, Oct 1993.

[20] Leon Gerritsen. Predicting student performance with neural networks. 05 2017.

[21] Frauke Günther and Stefan Fritsch. neuralnet: Training of neural networks. *The R journal*, 2(1):30–38, 2010.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[23] David J Hand. Principles of data mining. *Drug safety*, 30(7):621–622, 2007.

[24] Dorina Kabakchieva. Predicting student performance by using data mining methods for classification. 13, 03 2013.

[25] Dorina Kabakchieva. Predicting student performance by using data mining methods for classification. *Cybernetics and information technologies*, 13(1):61–72, 2013.

[26] Zlatko Kovacic. Early prediction of student success: Mining students' enrolment data. 2010.

[27] Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*, volume 453. Springer Science & Business Media, 1998.

[28] Hongjun Lu, Rudy Setiono, and Huan Liu. Effective data mining using neural networks. *IEEE transactions on knowledge and data engineering*, 8(6):957–961, 1996.

[29] Yishay Mansour. Pessimistic decision tree pruning based on tree size. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 195–201. Citeseer, 1997.

[30] John Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine learning*, 4(2):227–243, 1989.

[31] Lecturer Mrs. Bharati M. Ramageri. Data mining techniques and applications. 1, 08 2016.

[32] Sayyed Naqvi. Factors affecting students' performance a case of private colleges. 3, 02 2006.

[33] IES NCES. National center for education statistics. Retrieved from `https://nces.ed.gov/fastfacts/display.asp?id=372`.

[34] Joakim Nivre. Machine learning basic methodology. Retrieved from `http://stp.lingfil.uu.se/~nivre/gslt/method07ho.pdf`.

[35] Jill M. Norvilitis and Howard M. Reid. Predictors of academic and social success and psychological well-being in college students. 2012.

[36] Victor Oladokun, A T Adebanjo, B Sc, and O E Charles-Owaba. Predicting students academic performance using artificial neural network: A case study of an engineering course. 9, 04 0002.

[37] Edin Osmanbegović and Mirza Suljić. Data mining approach for predicting student performance. *Economic Review*, 10(1):3–12, 2012.

[38] Cristobal Romero, Pedro G Espejo, Amelia Zafra, Jose Raul Romero, and Sebastian Ventura. Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146, 2013.

[39] Cristóbal Romero and Sebastián Ventura. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010.

[40] Cristbal Romero and Sebastian Ventura. Educational data mining: A review of the state of the art. 40:601 – 618, 12 2010.

[41] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[42] Rahul Saxena. How decision tree algorithm works, 2017.

[43] Amirah Mohamed Shahiri, Wahidah Husain, et al. A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, 72:414–422, 2015.

[44] Raymund Sison and Masamichi Shimura. Student Modeling and Machine Learning. *International Journal of Artificial Intelligence in Education (IJAIED)*, 9:128–158, 1998.

[45] R Core Team. R language definition. *Vienna, Austria: R foundation for statistical computing*, 2000.

[46] Terry M Therneau, Elizabeth J Atkinson, et al. An introduction to recursive partitioning using the rpart routines, 1997.

[47] N VENGATESAN. Efficiency of decision trees in predicting students absentiesm in an academic year using c4. 5 algorithm.

[48] Muslihah Wook, Yuhanim Hani Yahaya, Norshahriah Wahab, Mohd Rizal Mohd Isa, Nor Fatimah Awang, and Hoo Yann Seong. Predicting ndum student's academic performance using data mining techniques. In *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on*, volume 2, pages 357–361. IEEE, 2009.

[49] Albert K. W. Wu and Chun H S Leung. Evaluating learning behavior of web-based training (wbt) using web log. In *ICCE*, 2002.

[50] Surjeet Kumar Yadav and Saurabh Pal. Data mining: A prediction for performance improvement of engineering students using classification. *arXiv preprint arXiv:1203.3832*, 2012.