UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

APPLICATION OF ARTIFICIAL NEURAL NETWORKS, GRADIENT BOOSTED DECISION TREES, AND MULTILEVEL LOGISTIC MODELS IN A SUPERVISED LEARNING ENVIRONMENT TO INVESTIGATE DIFFERENCES IN CLASSIFICATION PERFORMANCE WHEN PREDICTING COLLEGE ENROLLMENT

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

COLLIN CHRISTENSEN
Norman, Oklahoma
2018

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/299680

APPLICATION OF ARTIFICIAL NEURAL NETWORKS, GRADIENT BOOSTED DECISION TREES, AND MULTILEVEL LOGISTIC MODELS IN A SUPERVISED LEARNING ENVIRONMENT TO INVESTIGATE DIFFERENCES IN CLASSIFICATION PERFORMANCE WHEN PREDICTING COLLEGE ENROLLMENT

A DISSERTATION APPROVED FOR
THE DEPARTMENT OF PSYCHOLOGY

BY

_____
Dr. Robert Terry, Chair

_____
Dr. Eric Day

_____
Dr. Jorge Mendoza

_____
Dr. Hairong Song

_____
Dr. Maeghan Hennessey

© Copyright 2018 by COLLIN CHRISTENSEN
All Rights Reserved.

To Ashley.
Boomer.

ACKNOWLEDGMENTS

The following dissertation would not have been possible without involvement from many people. I would like to thank my wife, Ashley, for endless love, support, encouragement, patience, and understanding while I worked on this dissertation. My children, Knox, Nash, & Silas, for the unexpected, but much appreciated interruptions. My parents, Chuck and Raelee, for showing me how to appreciate education, and persevere until the job is done. My major professor, Dr. Robert Terry, for unforgettable wisdom, motivation, challenges, laughs, and friendship. My committee, Dr. Eric Day, Dr. Jorge Mendoza, Dr. Hairong Song, and Dr. Maeghan Hennessey for partnering with me on this endeavor. This process has been an unforgettable journey, and I'm so fortunate to have experienced it with all of you.

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv
TABLE OF CONTENTS v
LIST OF TABLES vii
LIST OF ILLUSTRATIONS viii
LIST OF EQUATIONS viii
ABSTRACT ix
CHAPTER I: INTRODUCTION 1
CHAPTER II: LITERATURE REVIEW 5
2.1 Description of Data Mining 5
2.2 CRISP-DM Methodology 7
2.3 Data Mining in Education 9
CHAPTER III: METHODOLOGICAL REVIEW 12
3.1 Classificaiton vs Prediction 12
3.2 Artificial Neural Networks 13
3.3 Boosted Decision Trees & Gradient Boosting 18
3.4 Multilevel Logistic Regression 23
3.5 Model Comparison 25
3.6 Summary 29
CHAPTER IV: EXPLANATION OF DATA AND VARIABLES 30
4.1 Data Procurement 30
4.2 Data & Variables 31
4.3 Data Preparation 35
4.4 Data Usage 37
CHAPTER V: METHODLOGY 39
5.1 Overview 39
5.2 Participants 39
5.3 Data Processing 40
5.4 Descriptive Statistics 42
5.5 Procedure 47
5.5.1 Gradient Boosted Decision Trees 50
5.5.2 Artificial Neural Networks 54
5.5.3 Multilevel Logistic Regression 56
5.5.4 Model Comparison 57
5.6 Summary 58
CHAPTER VI: RESULTS 59
6.1 Overview 59
6.2 Gradient Boosted Decision Trees Model Results 59
6.2.1 Gradient Boosted Decision Trees - Grade 9 59
6.2.2 Gradient Boosted Decision Trees - Grade 10 60
6.2.3 Gradient Boosted Decision Trees - Grade 11 62
6.2.4 Gradient Boosted Decision Trees - Grade 12 63
6.3 Artificial Neural Networks Model Results 65
6.3.1 Artificial Neural Networks - Grade 9 65
6.3.2 Artificial Neural Networks - Grade 10 67

6.3.3 Artificial Neural Networks - Grade 11 68
6.3.4 Artificial Neural Networks - Grade 12 70
6.4 Multilevel Logistic Regression Model Results 71
6.4.1 Multilevel Logistic Regression - Grade 9 71
6.4.2 Multilevel Logistic Regression - Grade 10 73
6.4.3 Multilevel Logistic Regression - Grade 11 74
6.4.4 Multilevel Logistic Regression - Grade 12 76
6.5 Model Comparison 65
6.5.1 Detailed Model Results 79
6.5.2 Summary 84
CHAPTER VII: DISCUSSION 86
7.1 Overview 86
7.2 Discussion of Primary Findings 86
7.3 Discussion of Secondary Findings 87
7.4 Ensemble Models 88
7.5 Future Direction 89
REFERENCES 91
APPENDIX 1: Summary of Variables 102
APPENDIX 2: Artificial Neural Network Grade 9 Node Weights 103
APPENDIX 3: Artificial Neural Network Grade 10 Node Weights 108
APPENDIX 4: Artificial Neural Network Grade 11 Node Weights 113
APPENDIX 5: Artificial Neural Network Grade 12 Node Weights 120
APPENDIX 6: Model ROC AUC Estimates by Grade 127

LIST OF TABLES

Table 1: Descriptive Statistics for Grade Point Average by Grade 43
Table 2: Descriptive Statistics for Attendance Variables by Grade 44
Table 3: Descriptive Statistics for Behavioral Variables by Grade 45
Table 4: Descriptive Statistics for Aggreagated Grade 12 Data 46
Table 5: Descriptive Statistics for School Level Homeless, Special Education, and Free/Reduced Lunch Variables 46
Table 6: Descriptive Statistics for School Level Turnover, Transfer within District, and Dropout Variables 48
Table 7: Grade 9 Gradient Boosted Decision Tree Predictor Importance 60
Table 8: Variables Used in Model Development 101
Table 9: Grade 9 Gradient Boosted Decision Tree Model Performance Metrics 60
Table 10: Grade 10 Gradient Boosted Decision Tree Predictor Importance 61
Table 11: Grade 10 Gradient Boosted Decision Tree Model Performance Metrics 62
Table 12: Grade 11 Gradient Boosted Decision Tree Predictor Importance 63
Table 13: Grade 11 Gradient Boosted Decision Tree Model Performance Metrics 63
Table 14: Grade 12 Gradient Boosted Decision Tree Predictor Importance 64
Table 15: Grade 12 Gradient Boosted Decision Tree Model Performance Metrics 65
Table 16: Grade 9 Artificial Neural Network Global Sensitivity Analysis 66
Table 17: Grade 9 Artificial Neural Network Performance Metrics 67
Table 18: Grade 10 Artificial Neural Network Global Sensitivity Analysis 68
Table 19: Grade 10 Artificial Neural Network Performance Metrics 68
Table 20: Grade 11 Artificial Neural Network Global Sensitivity Analysis 69
Table 21: Grade 11 Artificial Neural Network Performance Metrics 70
Table 22: Grade 12 Artificial Neural Network Global Sensitivity Analysis 71
Table 23: Grade 12 Artificial Neural Network Performance Metrics 71
Table 24: Grade 9 Multilevel Logistic Regression Results 72
Table 25: Grade 9 Multilevel Logistic Regression Performance Metrics 73
Table 26: Grade 10 Multilevel Logistic Regression Results 74
Table 27: Grade 10 Multilevel Logistic Regression Performance Metrics 74
Table 28: Grade 11 Multilevel Logistic Regression Results 75
Table 29: Grade 11 Multilevel Logistic Regression Performance Metrics 75
Table 30: Grade 12 Multilevel Logistic Regression Results 76
Table 31: Grade 12 Multilevel Logistic Regression Performance Metrics 77
Table 32: Grade & Model Level Performance Metrics 78
Table 33: Artificial Neural Network Grade 9 Node Weights 103
Table 34: Artificial Neural Network Grade 10 Node Weights 108
Table 35: Artificial Neural Network Grade 11 Node Weights 113
Table 36: Artificial Neural Network Grade 12 Node Weights 120
Table 37: Model ROC AUC Estimates by Grade 127

LIST OF FIGURES

Figure 1: Artificial Neural Network Sample Structure 14
Figure 2: GBDT Performance Metrics by Grade Level 81
Figure 3: ANN Performance Metrics by Grade Level 82
Figure 4: MLR Performance Metrics by Grade Level 83
Figure 5: MCC Performance Scores by Grade and Level 84

LIST OF EQUATONS

Equation 1: Basic Multi-layer Artificial Neural Network Function 15
Equation 2: Artificial Neural Network Single Node Error Term 17
Equation 3: CART Minimization Function for Variable Selection 20
Equation 4: Within Node Deviance Function 20
Equation 5: Decision Tree Node Fit 20
Equation 6: Multilevel Regression – Level 2 Function 24
Equation 7: Multilevel Regression – Level 1 Function 25
Equation 8: Model Accuracy 27
Equation 9: Model Sensitivity 27
Equation 10: Model Specificity 27
Equation 11: Model Precision 28
Equation 12: Matthews Correlation Coefficient 28
Equation 13: Average Multinomial Deviance for Boosted Trees 52
Equation 14: Global Variable Importance for Boosted Trees 53
Equation 15: Global Variable Importance with Unequal Classification Correction 53
Equation 16: Artificial Neural Network Node Deviance 55
Equation 17: Iterative Reduction of Deviance Function 55
Equation 18: Global Sensitivity Measure 55
Equation 19: Grade 9 Multilevel Logistic Regression Model 72
Equation 20: Grade 10 Multilevel Logistic Regression Model 73
Equation 21: Grade 11 Multilevel Logistic Regression Model 75
Equation 22: Grade 12 Multilevel Logistic Regression Model 76

ABSTRACT

The use of data mining algorithms for applied practice is becoming commonplace in many industries. The application of these models to the domain of educational data and practice could provide significant gains in understanding and implementation of prediction in the classroom. The wealth of data collected from students as they progress through a traditional education track could benefit greatly from machine learning and data mining. The present dissertation is designed to examine the usefulness, when compared to Multilevel Logistic Regression, of Artificial Neural Networks and Gradient Boosted Decision Trees, at predicting college enrollment using data collected as students progressed through high school. Because of the immense amount of data that data mining algorithms can interact with, the emphasis is placed on, but not limited to, variables representing difficulty of coursework, advanced placement, STEM vs non-STEM, behavioral referrals, attendance, and any statewide standardized testing. The grade level data was analyzed independently for each model to determine at what pace model predictive consistency increased as new and more relevant information was collected. The comparison of model predictive capacity revealed that certain data mining algorithms could indeed be used in place of traditional statistical models, but the gains were not always consistent across all grade levels. Implications and future research are discussed.

CHAPTER I: INTRODUCTION

In recent years, many academic and applied fields have seen an onset of data mining & machine learning techniques being implemented into standard protocol (Han & Kamer, 2011). The emergence of large-scale, automated data collection combined with the new methods of making large data available has established a need for machine learning algorithms to parse through vast amounts of data. This marriage of technological advancements and the operationalization of computers in most daily activities has not only created an abundance of data, but also allowed for a greater body of available data across most domains.

Fields such as education, computer science, finance, health sciences, production, and business have found ways to utilize data mining techniques for data extraction, data cleaning, and pattern recognition ultimately leading to faster, more efficient decision making (Hastie, Tibshirani, & Friedman, 2009). With this abundance of large datasets, systems of analytic techniques and exploratory methods become a necessity to organize and display information in an intelligible, meaningful manner. This is the primary reason data mining has been extended beyond an available option and become a necessity in many instances. Many data mining implementations being used in large corporations also contain automated machine learning functions. These algorithms perform analytics and decision/solution recommendation, but also have the capability to continually re-deploy analyses with every new data point gathered (Witten & Eibe, 2005). This allows the user to spend less time on optimization and maintenance of an analytic environment, while also permitting the machine component to continually

recalibrate weights for more optimized, relevant predictions. This continuous process also allows the models to adapt to the data as the data might change (Stoean, Pruess, Stoean, El-Darzi & Dumitrescu, 2009).

The unique approach to model development when viewed with an abundance of data is also one of the primary contributors that differentiates classical statistics from data mining and machine learning. The massive amounts of data becoming available in modern day systems allows for a much more exploratory approach to be implemented. Many data mining models place an emphasis on utilizing large quantities of data that, in many cases, could not be handled easily by common statistical techniques. Due to this strength of data mining models, it is common in data mining methodology to emphasize greatly understanding the domain and data so to not generalize and over-fit a prediction model (Lavrac, 1999). Hastie et al. (2009) stated that the type of learning being done in data mining referred broadly "to approaches that take a more inductive approach to building a model, allowing the data a greater role in suggesting the correct relationship between variables rather than imposing them a priori."

Specifically, in the realm of education, data mining is being implemented in unique situations, but not yet widespread in its application. One area of improved usage is in the measurement of unique student models for student classification (Ayala & Yano, 2009). The onset of data being gathered will now allow for unique models to be built at the student level, so learning systems can become more custom fit for individuals rather than clustered groups. Baker & Yacef (2009) anticipated that with the access and organization of large amounts of student level data,

machine learning algorithms can now be implemented to track and model students' knowledge, motivation, disposition, as well as, many personality traits that impact a student's educational journey. The move towards more machine-based assessments and computer adaptive measurement profiles is a by-product of a shift toward a more digital learning environment. Rupp, Nugent, & Nelson (2012) proposed that assessments are moving away from fixed-form stand-alone tests combined with short-form responses to robust adaptive assessment suites composed of performance-based tasks administered collaboratively in digital learning environments.

Given the large amount of data being collected throughout a students' academic progress, along with the standardized testing batteries being implemented at many milestones in a student's academic career, the field of educational research has become inundated with data that is either not being utilized to its full advantage or not being used in conjunction with other important fields (Murtaugh, Burns, & Schuster, 1999). By adapting the machine learning algorithms developed for data mining in other domains to the realm of educational research, new pattern detection approaches can assist in sifting through large amounts of data (Cristianini & Shawe-Taylor, 200).

The current study centers around taking advantage of a multitude of educational data and seeking out reproducible patterns to enrich prediction of college enrollment. This was achieved by examining large amounts of data covering many domains of a student's life and experience, and programmatically parsing

through it for identifiers indicating a higher probability of involvement in higher education.

The focus of this dissertation is not to prove machine learning algorithms have a place in educational research, because data mining has already impacted many facets of our national education system (Bhise, Thorat, & Supekar, 2013). The focus is instead to compare the predictive efficacy of the most commonly used machine learning algorithms when applied to the academic data commonly collected by state institutions. The uniqueness of this dissertation resides not only in the comparison of advanced methodology with more standardized statistical methods, but also the significance and breadth of the data collected for the analyses. A primary component of analysis will be a comparison to a more traditional statistical technique in use with most academic research. This measure is not intended to act as a baseline, but instead, one component of a general, unbiased comparison between prediction models.

The data being utilized for these analyses are significant due to, not only, the extended duration in which data collection took place (high school grades through early college years), but also the collection of multiple cohort years (3 separate cohorts) to control for any dependencies that could exist due to events occurring in a single academic year. This primed the data for a suitable comparative study, examining the predictive accuracy of the three models that will be detailed in a later section.

CHAPTER II: LITERATURE REVIEW

2.1 - Description of Data Mining

Data mining carries many definitions based on what type of professional is using it, as well as, the reason in which it is being used. The realm of computer science would view data mining through a different lens than a marketing researcher. Many of the definitions vary in terms of the amount of computational prowess versus the statistical methodology (Quinlan, 1986; Quinlan, 1993). When data mining began, computer scientists would primarily label it pattern detection using a series of algorithms, while the market researcher would view it as an analytic tool based more heavily in statistics than computer science (Shute, 1993). Pregibon (1996) provides one of the most universal definitions of data mining by stating that it is composed of three parts: statistics, artificial intelligence, and database systems research. The extensiveness and generality of the definition is due, in part, to the many tools and techniques that all reside within the scope of data mining as a field. There are many data management and exploratory methods that would rely much more on the database research portion of the definition. In turn, there are classification tools that would rely much more heavily on the statistical portion of the definition. Overall, data mining is best described as a pseudo-automatic process by which potentially hidden patterns and relationships in information are discovered (Dorian, 1999).

Gorunescu (2011) states that data mining has three "generic roots" that make up the field. The first, and oldest, root is statistics. Statistics provides well-researched techniques, such as exploratory data analysis (EDA), that identify

pattern-oriented relationships between bodies of variables when there is no information about the nature of the variables (Tukey, 1977). The second "root", artificial intelligence, is much more recent in origin than statistics. Artificial intelligence takes a heuristic approach to problem solving, contributing information processing techniques to the data mining procedure (Gorunescu, 2011). Artificial intelligence is commonly labeled as Machine Learning in most applied analytic environments. The third "root" is database systems research. This is made up of techniques such as data acquisition, data cleaning, and data management (sub-setting, creation of new variables from multiple variables' information, etc.), and provides the basis from which the information is mined (Witten & Eibe, 2005).

Data mining can be dissected into two primary areas, similar to statistics, predictive objectives (e.g. continuous outcome prediction, classification, anomaly detection) and descriptive objectives (clustering, visual exploration, association rule development, sequential pattern detection). Within these two areas, many methods and practices exist surrounding key aspects of pattern detection and outcome prediction. The many models and algorithms available under the umbrella of data mining can be viewed as tools that the professional must implement based on the uniqueness of the data and desired outcome. One common trait in data mining, that spans across multiple concentrations and multiple theoretical backgrounds, is the idea of data mining as a series of very important steps that must be completed in a very strict order. Most institutions and organizations that utilize data mining recommend a specific methodology known as the Cross Industry Standard Process

for Data Mining (CRISP-DM), but many areas use a modified version of CRISP-DM, usually merging or splitting the steps (Shearer, 2000).

2.2 CRISP – DM Methodology

CRISP-DM is constructed of six equally important steps. Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. During the Business Understanding step, the primary goal is to identify the project objectives. These objectives could include, but are not limited to, success criteria for tools or techniques, risks and contingencies, and project plan outcomes (Chapman et al., 2000).

The Data Understanding step involves collecting and reviewing the data. This could involve creating descriptive reports on data and reviewing the collection process, exploring the data, and data quality verification. The Data Preparation step includes selecting and cleaning the data. During this step, it is important to describe the rationale for including or excluding the data, creates reports describing the data cleaning methods utilized, create the analyzable datasets (subsetting, transposing, merging variables, etc.), and reformat variables to prepare for analysis (Chapman et al., 2000). This step is important to approach very carefully, because the orientation, format, and type of data might not be ready for the modeling step. If data cannot be analyzed properly, then the whole data mining process could provide improper or inaccurate results.

The Modeling step contains most of the statistical techniques and assessments. The goal of this step is draw conclusions related to your goals set in the Business Understanding step. This could include, but is not limited to, creating

models, classifications, predictive measures, and assessment of the predictive measures implemented (Clifton & Thuraisingham, 2001). Model assessment is also a very important part of the Model step. Assessing the fit of parameters and revising parameters included in the model are both vital actions when building a model. This step usually consists of making judgments on the success of models when compared with each other, basing model assessment on the accuracy and appropriateness of model fit (Chapman et al., 2000).

The Evaluation step is similar to the assessment portion of the Model step. The model assessment being done in the previous step assessed the accuracy and generality of a model, while the Evaluation step assesses the degree to which the model meets the criteria established during the Business and Data understanding phases (Leaper, 2000). It is not until a model possesses good fit, and satisfies the standards and goals set forth by the researcher that it is accepted as an appropriate model. The final step is the Deployment step. This step is vital when utilizing data mining for the development of solutions in an applied setting. It involves applying the results of the data mining procedures and monitoring these changes to ensure that the proper decisions were made (Chapman et al., 2000). This study will not involve the use of the Deployment step. A future direction component will follow the final evaluation step, as this study is not designed in a way in which the conclusions could be brought into action.

These steps outline the basis for data mining and its ability to be implemented. Although the exact structure of CRISP-DM cannot be fully implemented in this analysis, due to the nature of the project, the framework was

utilized as closely as possible. The primary component that will guide this study is the utilization of supervised learning. Supervised learning is required in situations where there is no unique fit measure or acceptance test available for the utilized models. Supervised learning techniques incorporate every input variable into the initial analysis called the training model. This model is developed on only a portion of the data and done in such a way that the learning algorithm being used seeks suitable functions that relate the input variables and output variables. This allows the algorithm to see the input and output data simultaneously to develop a model that represents the relationship between the two. Where this practice differs from methods incorporated in traditional statistics is the model selection, error reduction, and input removal that takes place. Most data mining algorithms will train their model by creating hundreds, if not thousands, of unique models, testing them all, then selecting the model or models that recreate the data the best. Machine learning algorithms like artificial neural networks even back propagate during the modeling phase (Han, Kamber, & Pei, 2012). This allows the algorithm to move forwards and backwards through the series of input and output variables to iteratively test and retest weights applied, thus removing error with each estimate (Rumelhart, Hinton, & Williams, 1986). This will be discussed in greater detail at a later point in the study.

2.3 Data Mining in Education

Educational measurement has experienced a shift in focus from traditional graduation rates, to more attention focused on college readiness (Strauss & Volkwein, 2004). As the reality of an increase in students attending college or

university becomes more evident, it is vital to accurately measure how prepared students are for attending post-secondary school (Birnie-Lefcovitch & 2000). Although consensus agrees that college readiness is vital to understand, there still exists many opinions as to what factors actually contribute to college readiness (Conley, 2007). Desjardins & Lindsay (2008) state that in most cases, some combination of actual quantitative measurables (e.g. GPA, count of advanced courses taken, etc.) and designed assessments geared toward post-secondary school achievement provide valuable information for predicting college readiness. Data mining models provide a new set of tools to better investigate the many patterns that exist within educational data.

Due to the financial implications involved, data mining models are more commonly being implemented for predicting student enrollment in college, attrition due to intermittent circumstances, and key motivators university administration can control concerning enrollment expectations (Luan & Zhao, 2006; Brewe, Kramer & O'Brein, 2009; Delen, 2010; Herzog, 2006). There is also research taking place in areas with less financial impact on institutions. predicting academic differences that exist for distance learning students, focusing resources towards non-traditional students to lessen academic churn, and locating trends in drop-out and retention fluctuations of specific student type (Kotsiantis, Pierrakeas, & Pintelas 2004; Siraj & Abdoulah, 2009; Herrera, 2006).

It has not taken long for the practice of data mining and machine learning to become functional in the field of education, and this trend will only grow as more methodology and application become proven with research and practice. The

overarching methodology of data mining and the practices of data mining within the realm of education have been described. The primary focus of this study is to provide a framework and comparison for how these models interact with a traditional statistical model when viewing large-scale educational data. The focus will now turn to the specific models being implemented within the practice of data mining and machine learning.

CHAPTER III: METHODOLOGICAL REVIEW

3.1 Classification vs Prediction

Modeling with predictive data mining models can generate two primary outcomes, classification and prediction, principally determined by the data being analyzed and the model being implemented (Weiss, Kulikowski, 1991). The identification and purpose of the two model types is based on the format of the outcome variable being predicted and the unique needs that accompany the data being investigated. Classification describes the process of creating a function distinguishing data into various classes or levels. The outcome variable for a classification model is always discrete and unordered. In contrast, prediction models do not classify into levels or categories, but instead model outputs made up of continuous outcomes, similar to multiple regression (Han, Kamber, and Pei, 2012). Prediction models are implemented to predict numerical data values rather than the discrete categories present in the classification output. Data mining in applied applications even allows the model to decide the proper outcome for prediction and form fit the model to best represent patterns accompanying that prediction.

Many types of data mining models (e.g. Classification and Regression Trees & Artificial Neural Networks) have the ability to perform as classification models and regressions models, with most models also allowing for both types of variable to be present as predictors (Alpaydin, 2011). Since one of the primary purposes of data mining as a practice is to detect reproducible patterns in data large enough that the signal is hardly detectable when compared to the noise, it is vital that the appropriate tool from the data mining toolbox is selected for the data.

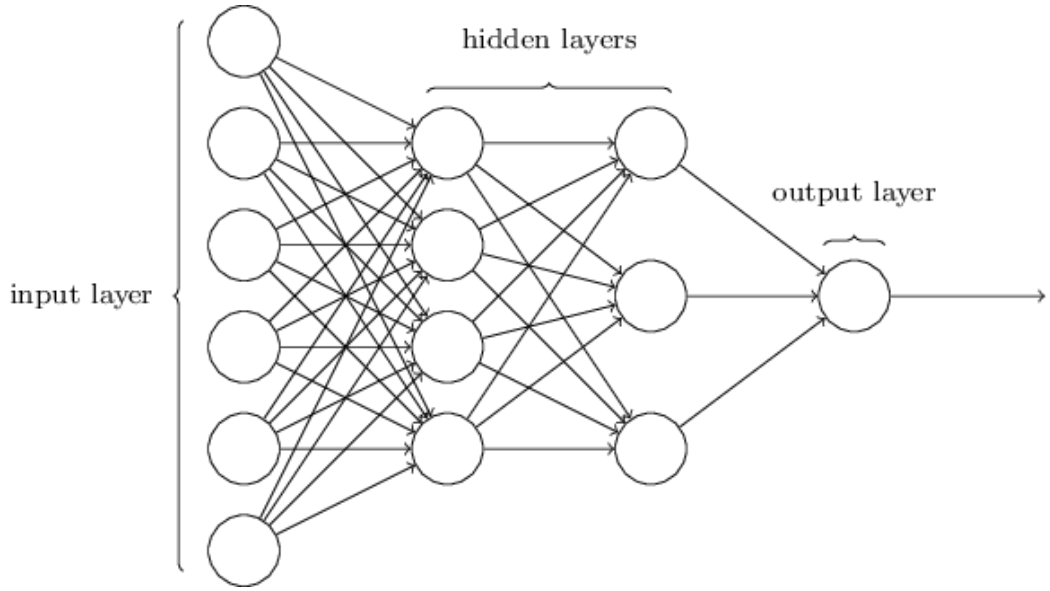
The models selected for comparison in this study were selected based on the following criteria: presence in current research and experimentation, availability of software required for implementation, and the models most utilized in current applied practice. The two primary machine learning models selected to characterize data mining are Artificial Neural Networks and Boosted Trees (Gradient Boosted Regression Trees). To provide a baseline for comparison with more classically utilized statistics, Multilevel Logistic Regression will also be utilized to analyze the data.

3.2 Artificial Neural Networks

Cheng and Titterington (1994) summarized Artificial Neural Networks (ANNs) as "the mathematical models represented by a collection of simple computational units interlinked by a system of connections." ANNs can be viewed as a complex system of nonlinear relationships composed of hidden layers and intuitive learning mechanisms (Taylor, 1999). Hastie, et al. (1999) described ANNs as "A two-stage regression or classification model… [in which] the central idea is to extract linear combinations of the inputs as derived features, and then model the target as a nonlinear function of these features." The use of ANN models allows for processing of many units of data that, when viewed together, seek out trends and relationships between input and output variables (Sibanda & Pretorius, 2012). Haykin (2008) described ANNS as a "biologically inspired analytical technique, capable of modeling extremely complex non-linear functions." The biologically inspired component that Haykin mentioned comes from the architecture of the network when viewed from input to outcome. The ANN structure consists of

processing nodes that exist as inputs for the model, joined with "neurons" that are interconnected through groups of weights, similar to the synaptic connections located throughout the nervous system (Mehrotra, Mohan & Ranka, 1997). This architecture allows for a "signal" to flow from input to weights to output and in reverse order. In between each layer of the ANN, a complex set of nonlinear models communicate information through the layers until convergence occurs in the output layer (Bishop, 1995; Bishop, 2006). An image representing a sample ANN is presented in Figure 1 to show the structure of nodes and their relationships.

Figure 1: Artificial Neural Network Sample Structure



Within the ANN portrayed above, the input layer consists of the input variables included in the dataset, with each input variable representing a separate input neuron (Haykin, 2005). The hidden layers are created by the model to apply parameter weights to the layers of inputs (Singh, Parhar & Malla, 2015). Each input neuron can communicate with each hidden layer in a unique way, and any number

of input neurons can impact any number of hidden nodes within the hidden layers (Bishop, 1995; Funahashi & Nakamura, 1993; Webb, 1994).

When referring to the components of an ANN, the input variable or "input layer" is anything that is bringing some data or information into the model (Nowlan & Hinton, 1992). On the opposite end of the model, anything that holds a predicted value or weight is considered an "output layer." There are also transformation functions nested in between the input and output layers that are called "hidden layers" (Luger & Stubblefield, 1993). As data flows from the input layer, through the hidden layers, and then continues on through the output layers, weights are assigned to each interconnecting line between two nodes (Sietsma & Dow, 1991). When the data reaches a hidden layer, the node (neuron) aggregates all values arriving from the input layer and the overall input values are applied to the model. Then the information is output to the next layer, where new weights are calculated and the process starts again (Sibanda & Pretorius, 2012; Neal, 1996).

Most applied ANN models are multi-layer network, which allow multiple inputs to be mapped to hidden nodes and output nodes with a series of complex non-linear relationships (Hastie et al., 2009). The basic function for a multi-layer ANN is:

$$a_k = g_k\left(b_k + \sum_j g_j\left(b_j + \sum_i a_i w_{ij}\right) w_{jk}\right)$$

Equation 1

where $a_k$is the output node, g_k and g_j are the activation functions that will change based on type of prediction being made (regression or classification), and b is the bias (and weight decay if chosen to be included in the model). Bishop (1995)

recommends usage of nonlinear transformations in the hidden layers (e.g. tanh, sigmoid, logistic), due to the fact that multiple layers of linear transformations can be formulated in a single layer of computation fairly easily, and the primary goal is to take full advantage of the computational strengths of the network.

It is not a requirement that an ANN model only have one layer of hidden nodes (Bishop, 2006). The more layers within a network that exist allow for more levels of unique analysis containing the information carried from the previous layer. The issue of overfitting arises if networks become too complex and contain many levels of hidden nodes (Bishop, 1995; Ripley, 1996). When information is processed using an ANN, the network pushes the weights and bias obtained from the previous layer into the next node, allowing the algorithmic learning process to begin using the information processed in the previous layer (Mackay, 1995). This leads to very complex learning processes, especially once back-propagation is introduced and the data can flow back through the layers of the network (Neal, 1996). Models can be built with thresholds called weight decays from a programmatic standpoint, these will be discussed more at a later point. Back propagation and the bi-directional application in neural networks will be discussed in the next paragraph (Opper & Winther, 2000; Haykin, 2005).

The most commonly used ANNs can be classified into two categories, Feed Forward Neural Networks & Recurrent Neural Networks (Van de Cruys, 2014; Bishop, 2006). The main difference between how these two types of ANNs view data, is that Feed Forward Networks are not bi-directional, indicating a linear flow of data propagation from input variable to output variable. Recurrent Neural

Networks are bi-directional and allow for propagation of networks from later stages to earlier stages (Bitzer & Kiebel, 2012). When viewing Recurrent Neural Networks, the most common estimation method is the Multi-Layer Perceptron (MLP) network. This network consists of at least 2 layers of nodes (neurons), input layer, output layer, and possibly hidden layers. MLPs are unique due to the way the model compares the output variables with known outcome values to calculate and apply a more accurate value of predefined error (Olden & Jackson, 2002). This error for the most basic neural network can be viewed as the following:

$$E = 0.5(o - t)^2$$

Equation 2

where the error, *E,* is a function of the output value, *o,* and the target value, *t.* Once calculated, this error value is passed back through the network and adjusts the weights that have been applied to the models accordingly. This iterative process, called back-propagation, continues until a reduction in the overall error function is detected (Calcagno et. al. 2010). MLP is commonly accepted as the most functional and utilized ANN model due to the model's ability to learn and re-estimate very quickly on large bodies of data. Research done by Hornik (1990) revealed that when presented with an appropriate amount of arbitrary data, MLP models are capable of deriving highly unique non-linear arbitrary models at very high accuracy levels.

Similar to other data mining algorithms that exist within a supervised learning environment, ANNs must be trained during use and implementation (Nilsson, 1990). Supervised learning is the practice of splitting the data to train the model being developed on one portion of the data and test the model parameters to determine successful classification using the other portion of data (Hastie,

Tibshirani, & Friedman, 2009). Mentioned briefly in the last section, the most commonly used training method in ANNs is back-propagation. During the process of information flowing from the input layer to the output layer, back propagation occurs to enhance predictive accuracy. Back propagation allows for information to pass back through the ANN with an adjusted expectancy of the error function. This allows for the weights being applied to the data to be adjusted as the models learns more about how the various layers relate (Weir, 1991). The primary goal of utilizing ANNs with this approach is to train a network that will find the best combination of variables and weights that produce the least amount of error when validated against similar data (Han et al., 2012). Validation is typically performed by randomly splitting the data and testing the model outcome on a portion of the data that wasn't utilized for learning (Bishop, 1995). This validation method is also the most feasible method to use during this study due to the various types of models being implemented. When implementing non-comparable modeling techniques fit statistics commonly used in classical statistical theory are not applicable (Sietsma & Dow, 1991; Cawley & Talbot, 2007). This will be discussed in greater lengths during the methods section.

3.3 Boosted Decision Trees & Gradient Boosting

The second type of model that will be implemented in this study is the Boosted Tree model. The Boosted Tree model is a specific variation of the Classification & Regression Tree (CART) model, which is typically the basis used for comparison for all decision trees (King & Resick, 2014). CART is the body of algorithms utilized within the field of decision tree. Decision tree will be explored

first, followed by an explanation of boosting and various optimizations that can be deployed with CART.

Decision trees are non-parametric, supervised modeling algorithms that can repeatedly run checks to extract the highest valued information from a dataset, without manual intervention, when presented with a model containing some predictors (Crockett, Latham, & Whitton, 2017). As stated is the case with many data mining models, decision trees can exist with categorical predictors (classification trees) and continuous predictors (regression trees). Within the structure of the tree, there are root nodes, daughter nodes, and terminal nodes. The root nodes exist at the top of the tree and contain all of the data being used to build the model, the daughter nodes are the nodes that exist throughout the middle of the tree containing the algorithmically determined splits in the data, and the terminal nodes are the nodes at the bottom of the tree representing partitions in the data that cannot be split anymore (Breiman Friedman, Olshen, & Stone, 1984; Gorunescu, 2011).

CART models utilize recursive partitioning to fit non-linear relationships without any pre-processing or preparation of the data (Quinlan, 1986). Recursive partitioning collects all of the data in one node at the top of the tree and proceeds down creating splits in the data with additional nodes until the tree is fully formed (Strobl, Malley, & Tutz, 2009). The primary reasons that the partitioning ceases is either a lack of data or one of the stopping rules has been triggered. The splitting algorithm utilized in decision trees iterates through all predictor variables until the variable that creates the most unique separation in the sample is located (Friedman,

2001). The minimization function utilized to select which variable will be used for the split can be viewed as

$$min\left[min\sum(y_{i1}-c_1)^2+min\sum(y_{i2}-c_2)^2\right]$$

Equation 3

where y_{i1} is the value of the outcome variable in node 1, y_{i2} is the value of the outcome variable for node 2, c_1 is the predictor variable value of observation with membership to node 1, and c_2 is the predictor variable value of observation with membership to node 2 (Hastie et al., 2009; Quinlan, 1986).

The model attempts to locate splits that minimize the sum of squared difference between the values and the within node averages, then being summed across nodes that share a common parent node (Breiman, 2006). The greater the similarity two nodes' values have leads to smaller sum of squared difference values. The most common measure for this heterogenous, within-node value is expressed with the following deviance value:

$$D_m=-2\sum n_{ij}LNp_{ij}$$

Equation 4

where n_{ij}represents the total number of subjects from group I in node J, and p_{ij} represents the proportion of subjects from group I in node J (Elith, Leathwick, & Hastie, 2008). This deviance value will increase as within-node heterogeneity increases, thus indicating a lower level of strength in the prediction of the split (Breiman et al, 1986). One common representation of fit for all decision trees is:

$$D=\sum D_m$$

Equation 5

The iterative process of analyzing the impact of each variable continues once the resulting nodes are as homogenous as achievable when speaking of membership to one group or another. This node creation and "splitting" of the data continues until the within-node heterogeneity of the outcome cannot experience any greater reduction in the deviance of the data. As mentioned above, one of the reasons trees discontinue splitting into additional nodes are stopping rules that are deployed to stop the model from growing too large and losing too much accuracy (Dorian, 1999). As the tree grows too deep, there could exist too many splits in the data disallowing a justified amount of data to exist in each terminal node. While a shallower tree will lead to outcomes that are too heterogeneous (Han, Kamber, & Pei, 2012). These two instances are examples of the need for stopping rules such as pruning. Pruning is an integral component of CART modeling and allows for the tree to maintain accuracy and generalizability (Alpaydin, 2011). There are many pruning mechanisms in place based on what software is utilized for calculation, but at their root they all perform the same task and that is overgrowing the tree, then pruning the terminal nodes back to an optimal size.

Classically developed decision trees offer one rigid path of decisions that can be limiting in scope due to the focus being decided earlier in the model development process (Breiman et al., 1984). Due to increases in available software algorithms and computing power, decision trees have become much more versatile and less likely to hone in on one specific node split, causing the model to lose generalizability.

When Gradient Boosting is applied to the tree, creating a Boosted Tree, the concern of relying on a single rigid path is dissipated. The Gradient Boosting technique is an optimization technique that can be implemented for classification, regression, or rankings solutions (Brieman, 1998). Gradient Boosting leverages elements of Gradient Descent as well as Model Boosting. Gradient Descent being the process of minimizing an error function by moving in the opposite direction of the negative gradient (or residual), and Model Boosting being the process of adapting to a number of different loss functions with varying robustness to outliers (Freund & Schapire, 1996; Schapire & Freund, 2012). During this process, an ensemble, or additive, model is fitted in a forward step-wise progression. During each step, the model introduces what is called a "weak learner" that exists as a new weight that is meant to slightly improve on the weakest existing model component (Elith et al., 2012; Brieman, 1999).

The first successful boosting technique, Invent Adaboost, was implemented by Freund and Schapire (1997), but with a greater body of research, computational power, and data Gradient Boosting began to be developed in work by Friedman (2000). The "Gradient" component added to the algorithm was implemented to account for a large variety of loss functions (Friedman, Hastie, & Tibshirani, 2001). Gradient Boosting algorithms as they are used today, primarily compensate for residuals in a step-wise fashion so to continue reducing error by creating new nested regression or classification trees in ensemble models as the training data is analyzed (Friedman, 2002; Brieman, 1996).

3.4 Multilevel Logistic Regression

Multilevel modeling has historically been utilized to better predict outcomes in education related domains. This is due to the very natural hierarchy that comes about through the designation of students, schools, districts, regions, states, etc (Raudenbush & Byrk, 2012). The outcomes predicted by multilevel models can be oriented to focus on high levels and low levels within the data structure. When viewing the structure of a hierarchy, all levels that exist below a given level of the hierarchy are, by design, nested (Rocconi, 2013; Baeck & Van den Poel, 2012). These nested data structures will maintain high correlation with the structure that they are nested within. Due to this high correlation, regression models that assume independence of error and random sampling techniques are not appropriate (Singer, 1998). Standard errors can also be misestimated due to a failure to account for any dependence data might have on a higher-level structure within the hierarchy in which it is nested (Roberts, 2004). The assumptions that follow multilevel models account for the implicit relationship that exists between the levels of a nested hierarchy (Tabachnick & Fiddell, 2012).

Singer (1998) stated that one of the primary goals of utilizing multilevel modeling is to create functions of value at multiple levels of interest. When considering the current data set being analyzed, this would encompass functions directed at the school level, or second level, as well as, the student level, or first level. The data being utilized is limited to one school district, so there will be no need for a third level of hierarchy in the model implemented in this study.

The subsequent description of a use-case on the data collected for this study will aid in describing the relationship the levels of hierarchy have with the overall model. When examining the school level, one area of interest when utilizing multilevel modeling could be a categorical predictor for the academic level of a specific subject area offered at the school. In the data used for this study, consultation took place with the subject matter experts to grade, quantify, and standardize the "academic level" of the math, reading, and science courses offered at the various schools in the data set. After cleaning, this academic level field indicated if the math, science, or reading course being taken by the student (or offered at the school) was below the desired grade level, at the desired grade level, or above the desired grade level for the given school district.

If there is interest in modeling this at the school level to determine probability of college enrollment, the mean predicted probability can be portrayed as a combination of the grand mean predictor (Υ_{00}), the selected impact the aggregate academic level course taken at the school (Υ_{01}) has on the predict probability, the error associated with each individual school in the dataset (μ_{0j}), and the error associated with the individual students in each classroom (r_{ij}).

$$\Upsilon_{ij} = \Upsilon_{00} + \Upsilon_{01}(\text{Level}) + \mu_{0j} + r_{ij}$$

Equation 6

The model above represents the school level function. If there was interest in adding to the overall model by examining the impact of total AP credits earned on the predicted probability of college enrollment, the student level model would be viewed as:

$$\Upsilon_{ij} = B_{0j} + B_{1j}(\text{APCREDITHOURS})_{ij} + r_{ij}$$

Equation 7

with $B_{0j} = \Upsilon_{00} + \mu_{0j}$ and $B_{1j} = \Upsilon_{10} + \mu_{1j}$. Once all of the functions are combined, the overall model would allow for better understanding of the influence of variables at both levels of the hierarchy, along with the errors unique to school and students levels of the model. Along with the unique errors represented in each level of the model, it is also of interest to explain the variance captured in the slopes (τ_{11}), the variance captured in the intercepts (τ_{00}), and the covariance between the two (τ_{01}).

Multilevel modeling is not restricted to only prediction of continuous outcomes. This study is focused on predicting college enrollment, so the model will be applied to a dichotomous outcome. The basic principles of the model and its construction will still follow what was described above.

3.5 Model Comparison

The primary goal of this dissertation is to provide a comparison of the three models of interest within the context of large behavioral/educational datasets. The primary issue that arises when comparing the models' ability to properly select college enrollment is the method of which each model uses to depict optimization or success. Logistic regression computationally follows the primary constructs of traditional statistics, while the data mining algorithms both utilize supervised learning to measure model accuracy, sensitivity, specificity, and precision (Breiman et al., 1984). Due to there being no common ground between how these two models are traditionally interpreted, all three models will be compared using a supervised learning environment (Ludbrook, 2002; Suleiman, Tight, & Quinn, 2016).

As mention before, supervised learning takes place when a smaller portion of the data set is randomly sampled to "train" the model and decide parameters, then the model with updated parameters is applied to the rest of the data to assess how well it predicted the known outcomes (Fay, 2005). Since this is the natural process that would take place for artificial neural networks and gradient boosted tree algorithms, the primary modification will take place with the multilevel logistic regression models. The multilevel logistic regression models will be trained on the same subsample as the other two models, then tested against the remaining data to assess model fit.

The four major components of model fit, when working within supervised learning states, are model accuracy, model sensitivity, model specificity, and model precision (Brieman et al., 1984). To understand these three metrics, it is important to first understand the four possible outcomes that can occur with a categorical outcome. When trying to predict a dichotomous outcome like college enrollment, each observation of the test dataset can result in a True Positive (TP), False Positive (FP), True Negative (TN), or a False Negative (FN) (Jain & Zongker, 1997; Guyon & Elisseeff, 2003). A true positive would occur when the model correctly predicts a student enrolled in college. A false positive would occur when the model predicts a student will enroll in college, but that was not the outcome. A true negative would occur when the model correctly predicts a student not enrolling in college. A false negative would occur when the model predicts a student will not enroll in college, but that was not the outcome.

In gauging the overall Model Accuracy, or the model's ability to differentiate between students who would enroll and students who would not enroll, the proportion of true positives (TP) and true negatives (TN) from all evaluated observations must be calculated.

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

Equation 8

In gauging the overall Model Sensitivity, or the model's ability to determine college enrollment properly (ignoring successful prediction of students not enrolling in college), the proportion of students who were correctly predicted as enrolled, true positives (TP), from the total number of students who did enroll, true positive (TP) and false negative (FN) is calculated.

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

Equation 9

In gauging the overall Model Specificity, or the model's ability to properly predict students' who did not enroll in college (ignoring successful prediction of students enrolling in college), the proportion of students who were correctly predicted as not enrolling in college, true negative (TN), from the total number of students who did not enroll in college, true negative (TN) and false positive (FP), is calculated.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

Equation 10

Additional to the measure honing in on true positive rate (sensitivity) and true negative rate (specificity), it is important to also keep a measure of positive

predicted value, or Precision. This measure is captured as a ratio of true positives (TP) to true positives and false positives (FP).

$$\text{Precision} = \frac{TP}{TP+FP}$$

Equation 11

The final metric to consider when gauging the success of data mining algorithms is the Matthews Correlation Coefficient, or phi coefficient in some literature (Boughorbel, Jarray, & El-Anbari, 2017). This metric is commonly deployed when a machine learning model is attempting to measure the quality of a binary classification predicted from a supervised learning environment and is widely accepted as one of the supervised learning measures of fit that is least altered by an inconsistent classification ratio (Matthews, 1975). It gains value because of its ability to maintain balanced outcomes when the class sizes in the data are of drastically different sizes (Powers, 2011). On occasion, it becomes less valuable to only view accuracy, or the proportion of correct predictions, because the size difference between the two outcomes is drastically different (Perruchet & Peereman, 2004).

The MCC has a range of -1 to 1 where -1 indicates a fully incorrect binary classification, and 1 indicates a fully correct binary classification. The use of the MCC provides the most balanced gauge for how well classification models are performing. To calculate the MCC, it is necessary to utilize the prior calculations for true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Equation 12

It is common, when working with data mining algorithms, for the above described metrics to be included in a table referenced as a confusion matrix (Kohavi and Provost, 1998; Caelen, 2017). Many advanced models include components called confusion-matrix based attribute selection, which allows the model to automatically adapt model weights and continue calibration on new data based on values derived from the confusion matrix (Ming, 2011; Ruuska, Hämäläinen, Kajava, Mughal, Matilainen, & Mononen, 2018). For the purpose of this dissertation, the metrics will be utilized for reporting the best fitting models during the model comparison phase.

3.6 Summary

This section has provided an appropriate introduction to theories behind each of the three modeling techniques utilized in this dissertation. The section began by explaining classification versus prediction from a data mining stance. Next, the processes involved in the implementation of an artificial neural network, gradient boosted tree algorithm, and multilevel logistic regression model were described. Lastly, the metrics used to compare the models and the process for which they will be compared is described.

Now that the use and theory of the data mining algorithms and multilevel models have been described, a greater emphasis will be placed on explaining the primary methods employed for data acquisition and data preparation/cleaning.

CHAPTER IV: EXPLANATION OF DATA AND VARIABLES

4.1 Data Procurement

The data used for this study was acquired through and approved by work done with the University of Oklahoma. The high school level data, representing a large midwestern school system, was provided by the State Department of Education and was combined with the appropriate college level data from the corresponding state's higher education institutions. The joining of these datasets was completed by specialists designated through and approved by the University of Oklahoma, and multiple checks were put in place during the joining of the data so that purity was maintained. Required fields for matching were extended beyond just unique student ID, into such things as social security number, birthdate, birth year, and complete name. It was required that the two data files match on at least 80% of the matching fields or the observation was excluded.

The master dataset originally obtained for this study contained 32,435 variables and 19,728 observations, totaling approximately 640 million data points. This initial dataset was made up of 3 cohorts of students who attended school in the large midwestern school system. The data spans from 6^{th} grade through 12^{th} grade at the individual student level with fields for every recordable action throughout the students' academic career. The data also includes all alternative schools, magnet schools, and behavioral schools. Once the three cohorts were merged, the dataset was then combined with the corresponding college level dataset collected by the State Regents for Higher Education.

The methods implemented for matching the school level data with the college level data were validated internally by the State Regents for Higher Education and any non-matching cases were removed from the sample. The criteria for matching included social security number, first name, middle initial, last name, and date of birth. To be included in the dataset, each observation (student) was required to match on an established number of these criteria, all of which were set and validated by the State Regents for Higher Education before delivery of the data.

4.2 Data & Variables

As mentioned above, the dataset contained 32,435 variables, so each variable will not be explained in detail. This was due in part to how the data was collected and stored, creating a new variable for every possible grade level – record – student combination. The format of the variables and what was done to clean and process the data are discussed below in the Data Preparation section. Within the data, there were a number of naturally occurring subgroups. These subgroups were composed of Demographic fields, Academic fields, Behavioral fields, Social fields, and Enrollment/Attendance. The primary fields of interest from each of these subgroups will be explained below.

The Demographic subgroup contained sex, ethnicity, English as a Second Language (ESL), English Language Learners (ELL) resident status, homeless status, free/reduced lunch, special education classifications, physical impairment, other disability, and gifted & talented.

The Academic subgroup contained test scores such as EOI, CRT, OCCT, ACT, SAT, EXPLORE, and WIDA. Other Academic variables are GPA, Advanced

Placement courses taken, Advanced Placement credits earned, promotion, retention, benchmark test scores by subject, and credits earned. The EOI/OCCT variables were such things as content area of exam, raw and scaled scores, performance level, duration of student enrollment, English proficiency, and a flag for students taking the exam a second time due to unsatisfactory scores the first administration. Specific Academic variables were then created to add special focus to the analyses. A variable representing Cumulative Math GPA was calculated by coding GPA at the course level and only keeping math courses.

There was also a detailed effort to create and code variables representing the course level of the math and science courses offered. Subject matter experts associated with and approved by the University of Oklahoma, specializing in the courses being viewed, contributed information to this classification process based on course number at each respective school, as well as, the expected prerequisites for each course. They were classified as below expected course level (remedial), at expected course level, or above expected course level (AP, Honors, higher level math and science, concurrent enrollment) for each grade in the data. These variables were used to conceptualize the difficulty of the given math or science course in relationship to what the expected difficulty would be. These variables were created by first creating a list of every possible course name and course number from any public school included in the dataset (6^{th}-12^{th} grade). After the compilation of the course information, each course on the lists (one list for math courses and one for science courses) were coded in the given subject. The codes given represented one of the three groups mentioned above: below grade level, at grade level, or above

grade level, for each grade in the dataset. This information was then entered into the master dataset.

In calculating the STEM course GPA, a list of STEM course numbers were collected from the OSRHE. This list contained 109 courses that were deemed STEM courses by OSHRE. OSHRE was also responsible for denoting the coding scheme for Institution Type. This included Large State Research Universities (e.g. OU, OSU), Small State Colleges (e.g. UCO, SWOSU), and Community Colleges (e.g. OCCC, TCC). The calculation of the non-stem course GPA utilized the same list but controlled for the STEM courses and removed them from the aggregation.

The variables included in the Behavioral subgroup included things such as items from the EXPLORE test, items from the PLAN test, and items from the ENGAGE test. These items gauged things such as academic interest, willingness to put forth effort in school, details about post-high school plans, homework, family involvement in education, and perceived success. The Behavioral dataset also included variables on disciplinary action taken against the student. These variables included total number of referrals, type of referrals, total days suspended, reason for suspension, and number of truancies.

The variables included in the Social subgroup included primarily items collected from the ENGAGE test. This included items such as social connection with school personnel, managing goals, motivation, self-confidence, and determination to succeed. The variables in the Enrollment/Attendance subgroup included school attended, number of times primarily school changed during year, school at which state test was taken, time of entry at each school, time of exit at

each school, reason for exit, absences per school, total absences, days enrolled per school, total days enrolled, and attendance ratio per entire grade year. There were a number of higher-education variables that were included in the predictor side of the dataset, primarily as controls. Examples of these variables are type of higher-education institution, student status (full time or part time), and STEM GPA. The primary outcome variable of interest that was included with the higher-education data set was enrollment/retention within a higher education institution.

There was also a school level dataset created from the aggregation of student level data with the associated school code. This school level dataset was matched against statistics provided by the State Department of Education for each school for validation purposes. The aggregate values matched with a minimal expected amount of error due to uncontrollable factors such as students transferring to other schools during the school year, students being relocated to behaviorally centered programs during the year, and students experiencing multiple instances of extended out of school suspension or expulsion. It was determined that, for the purpose of this study, there was an acceptable amount of variance between the values reported on the state documentation and the actual aggregate values. Since the purpose of the study is predicting college enrollment, school level data that was negatively affected by students primarily experiencing behavioral interventions would be discernable in the Specificity metric of each model.

The second level dataset contained predominately academic, behavioral, and demographic variables representing the schools captured within the dataset. Examples of the variables that were included in this dataset are average student

GPA, average student test performance (e.g. EOI, ACT, SAT), average number of AP courses taken, average promotion/retention rate, average graduation rate, average attendance ratio, average number of suspensions, and average number of disciplinary referrals.

4.3 Data Preparation

The data preparation portion of the study was difficult due to the orientation of the dataset. The data was combined so that one student was represented by one observation line in the dataset. This was difficult primarily because grade level data were not separated into grade subgroups. Each student had a year of entry for each school and a numeric coding scheme made up of underscores and numbers that would place each test score, behavioral instance, course taken, etc. into a specific year within the student's academic career. Because of this coding scheme, a student's data under the variable titled GPA_2_2 (the first numeral representing the year and the second numeral representing the semester) that represented the students' cumulative GPA taken at the second semester of the 7th grade (because the 6th grade was the first grade in the dataset and the _2 represents the second year). Another student's data under the variable GPA_2_2 could represent the student's cumulative GPA taken at the second semester of the 11th grade (because the student transferred into the school system in the 10th grade and the _2 represents the student's second year in the dataset). This caused for additional coding to be written for each combination of every variable type and every grade/year combination.

Another aspect of the data cleaning effort was computing descriptive statistics and creating flags for variables containing cases that fell outside of the

defined range. There were 33 cases that contained variables with data that was out of range. Due to the importance of the variables that were out of range, the entire case was removed from analysis for each of these observations. After coding to recognize which variables placed students in grades, the primary dataset was split into seven grade level datasets. Each of these seven datasets contained one observation for each student who recorded data in that grade. Students who did not have recorded academic, behavioral, or higher education data were removed from the dataset. Due to how the datasets were constructed, students who did not have grade level data but still had an identifying number were removed from the dataset. Duplicate case filters were added to each grade to ensure each student was not improperly represented in a grade more than once. This process will be discussed at greater length during the next section.

Data reduction methods were also used to improve the analysis phase of the study. There were groups of variables removed that did not pertain to every dataset. The variables were mainly characterized by complete blocks of missing data, but the data dictionary was used to validate the removal. These variables were checked to ensure no data existed for the irrelevant grade level datasets, and then removed from the datasets. An example of this type of variable would be one containing an item from a standardized test administered in the 8^{th} grade, but due to the structure of the dataset, the variable existed at each grade level. These variables would appear completely missing for a student in the 12^{th} grade, because 12^{th} grade students did not take a test administered to 8^{th} grade students.

Additional data preparation tasks were creating dichotomous variables for outcomes of interest. A dichotomous code for whether or not the student enrolled in a higher education institution was created. This became the primary classification variable for all of the models. Another group of variables were created to represent whether or not a student had taken an AP course, as well as whether or not they obtained AP credit before graduation. The dataset already contained a variable for each AP test at each grade level, but there were so few data points in some of the AP testing groups, that the new variables represented whether or not the student received credit from *any* AP test, as well as a continuous variable representing how many AP courses they completed before graduation.

One of the largest blocks of data removed during the variable reduction phase was the ENGAGE test data. It was observed during the cleaning process that less than 0.7% of all observations contained any ENGAGE data at all. It was also discovered that only 0.5% of observations contained a complete ENGAGE test without missing observations. This was the case because the battery was only administered to a small number of students in one cohort. This was far too much missing data to include these variables in any analyses, so the blocks of variables representing the ENGAGE battery were removed.

4.4 Data Usage

The primary value being added by the dataset in use is its inclusiveness of all data collected from one entire large school district, joined together with all college enrollment data for the students. This allows for a complete representation of every student in the district from high school through the first year of higher education.

One of the leading strengths of most data mining models are their ability to adapt to, and gain value from, large datasets. The interest of this model comparison was rooted in the question of whether or not larger datasets lend more productivity to data mining algorithms over traditional statistical methods. Multilevel modeling has traditionally been a favored approach for analyzing educational data, but this study aims to investigate if predictive value gained from data mining models is greater than that of multilevel models with a large dataset.

CHAPTER V: METHODOLOGY

5.1 Overview

This chapter describes the data processing, modeling techniques, and model comparison methods that were utilized to compare the successfulness of models in predicting college enrollment. The following sections will discuss the process and justification of the participants, data processing method, and analytic implementations.

5.2 Participants

The data acquired for this dissertation contained three cohorts of students from a large Midwestern school district, with measures collected from the 6th grade through the 12th grade for each cohort. Upon receiving the master datafile, there were 32,435 variables and 19,728 observations. Once data was cleaned and merged, the final dataset contained 17,877 observations, with one observation matching each unique student ID. This data included students from any public school in the district, including magnet schools, behaviorally focused schools, alternative schools, and schools with focused special education programs. There are a total of 27 high schools in the data after magnet and alternative schools were included, although five of the schools recorded below 5 students per grade. The demographic nature of the school district is 51% Hispanic, 25% African-American, 15% Caucasian, 3% Native American, 2% Asian, and 4% who selected two or more racial/ethnic categories. Throughout all grades, approximately 1 in every 3 students are ESL/ELL with Spanish listed as their primary language.

5.3 Data Processing

The dataset used in this dissertation was created in the phases mentioned earlier during the Data Preparation section. Additional detail will be added in this section to describe the process utilized during some data processing decisions. SAS and SQL were utilized to manage the datafile in the earliest stages of the data processing phase. This was done to accommodate for the size of the file and it's need to be managed in an analytic environment. All fields and grades were managed at the student level to complete data cleaning and validation efforts. Once all data was cleaned, validated, and grades were joined across like unique student ID, a school level dataset was created via aggregation methods and validated against state reported data.

Due to the size of the master dataset, the data dictionary was used in place of descriptive analysis to initially partition the dataset into multiple portions. The data dictionary contained a brief description of name, variable type, and purpose for each of the 32,435 variables, along with a listing of each of the categorical response types for the appropriately structured variables. This was utilized to parse out variables of each type and begin sub-setting them into appropriate groups for easier consumption during the analysis. Partitioning of the data was done for a number of reasons, but primarily it was done to better allow the handling and cleaning of the data. The computers and software utilized during this dissertation could not process all of the raw data simultaneously, so two partitioning methods were used. First, data were separated by grade level after all cohorts were merged. This allowed for one master file per each grade across all of the data. The unique student ID's were analyzed for

duplicates within one grade, existing because a student was held back or required to take a specific grade over. When duplicate unique ID's existed, the observation with the most complete data fields, including fields indicating that the academic year was completed, were kept.

The amount of missing data that was compromised when student level data was aggregated across grades did not allow for analysis to take place at the level. The frequency of students transferring between schools, leaving the district, dropping out of school, and being dismissed from normal school activity via suspension, expulsion, or alternative school created a large number of missing fields. Students also, more frequently than expected, had demographic, school, and behavioral records, but no academic records. After examining the data dictionary, it was discovered that students who were enrolled in a grade level for an entire year, but were absent more than the allowed limit, failed the course, thus did not receive credit or a recorded grade point average. As with many models, if an observation has insufficient non-missing data, it will be removed. It was observed that the large number of missing fields that would be removed before analysis would weight the sample more in favor of those students enrolling in college, thus misestimating the weights for classification and altering the fit of the overall model. Due to this issue, the grades are being analyzed as snapshots.

While creating the school level dataset, all observations included in schools that had an insufficient number of data points to represent the school during multilevel modeling were removed from all models. This was done so that the data being trained and tested during the supervised learning process was the same for all

three models. There were four high schools removed from the data due to total student counts being below 5, with most of the schools reporting student body size as 1. Since the schools were coded with a numeric classifier rather than a school name, it was impossible to tell if these were behavioral programs/alternative schools or simply dirty data. More than half of the fields removed were also missing enough academic records that the student data would be removed from the model during processing. After the school level dataset was created for each grade, it was joined to the corresponding student level data.

Prior to the analysis, there were procedures in place to adequately split and validate the training dataset and the test dataset so the supervised learning model comparison could take place. This process will be better described at a later point in this section.

5.4 Descriptive Statistics

As mentioned in a previous section, the following metrics were calculated after data had been cleaned and duplicate unique student ID numbers had been removed. It is important to note that the variables displayed in this section were measured after aggregation had taken place across three cohorts, so there are not descriptive tables for each cohort included in the original data.

Table 1 displays the descriptive statistics for the grade point average variables calculated from the data. As explained in a previous section, each course number available at every High School was coded for its level in correspondence to the grade level of the student, as well as, the core subject area. During the preliminary research for the study, it was deemed useful to parse out STEM or

Science/Math course involvement from the total grade point average. Table 1 contains the observation count, mean, and standard deviation of standard grade point average at each grade level, and the observation count, mean, and standard deviation of math/science grade point average at each grade level. Since the analysis is being done by grade, each of the grades will be separated when descriptive values are provided.

Table 1: Descriptive Statistics for Grade Point Average by Grade

Variable	N	Mean (SD)
Overall GPA		
9th Grade	14,011	1.95 (1.16)
10th Grade	11,743	1.79 (1.39)
11th Grade	10,013	1.83 (1.26)
12th Grade	8,388	1.99 (1.24)
Math & Science GPA		
9th Grade	11,690	2.05 (1.09)
10th Grade	7,988	2.07 (1.11)
11th Grade	6,149	2.19 (1.08)
12th Grade	4,746	2.33 (1.01)

Table 2 displays the descriptive statistics for variables associated with the attendance metrics collected throughout the students' career. The average Days on Roll metric was calculated using start and end dates from each students' academic career by school/district code. These figures represent the disparity between days attended within the district and a full school year. The attendance rate was calculated using the total days on roll as a weight.

Table 2: Descriptive Statistics for Attendance Variables by Grade

Variable	N	Mean (SD)
Attendance Rate (Perfect Attendance = 1)		
9th Grade	14,011	.854 (.134)
10th Grade	11,743	.859 (.144)
11th Grade	10,013	.868 (.117)
12th Grade	8,388	.864 (.116)
Average Days on Roll (Max = 180)		
9th Grade	14,011	136.76 (52.64)
10th Grade	11,743	137.89 (50.42)
11th Grade	10,013	139.62 (47.33)
12th Grade	8,388	145.26 (41.56)

Table 3 displays the descriptive statistics for variables associated with the behavioral metrics collected on the students. These variables were created using disciplinary codes for each possible infraction that could take place on school grounds. The field representing referrals is measuring the number of reprimands the average student received during the grade listed. These referrals were filtered down to only series infractions leading to short-term or long-term suspension/expulsion. The average days suspended variable was created to represent the average length of punishment served for the referenced behavioral infractions. The average days suspended variable does include students who were expelled for an entire semester/year. This field represent all students receiving in-school/out-of-school suspension, or expulsion with a calculated, exact day count. If a student was removed from a school and did not record credits earned or a GPA, his/her record was removed from the analysis during the data cleaning phase.

Table 3: Descriptive Statistics for Behavioral Variables by Grade

Variable	N	Mean (SD)	Max
Average Number of Referrals per Student			
9th Grade	14,011	1.29 (2.67)	34
10th Grade	11,743	.987 (2.26)	22
11th Grade	10,013	.778 (1.99)	20
12th Grade	8,388	.536 (1.37)	15
Average Days Suspended per Student			
9th Grade	14,011	1.77 (7.76)	155
10th Grade	11,743	1.49 (8.03)	162
11th Grade	10,013	1.07 (5.84)	160
12th Grade	8,388	0.87 (7.58)	160

Table 4 displays the descriptive statistics for the district wide characteristics pulled from the aggregated data of 12th grade students. These variables were limited to reporting at the 12th grade due to the typical point in a student's career in which this information is collected. For all students completing the 12th grade, average ACT and SAT scores, average Advanced Placement credits earned, and remediation / Gifted & Talented statistics were calculated. A decimal value representing a ratio was utilized to depict the proportion or students requiring remedial math courses, requiring remedial science courses, and being involved in a gifted and talented program within the school. These were calculated utilizing flags created in the data that represented a student's involvement in a course number that was designated as remedial math, remedial science, or gifted and talented. Demographic variables were not reported on in this study because they were not utilized in the modelling of student's enrollment in higher education.

Table 4: Descriptive Statistics for Aggregated Grade 12 Data

Item	N	Mean (SD)	Min	Max
ACT Composite Score	2948	16.11 (3.78)	8.00	35.00
SAT Scores				
Reading	217	542.18 (101.55)	300.00	790.00
Math	217	516.93 (111.83)	290.00	760.00
Writing	217	520.56 (103.66)	200.00	800.00
AP Credits Earned	8388	0.04 (0.26)	0.00	5.00
Ratio of Students Requiring Remedial Math Course(s)	8388	0.79 (0.37)	0.00	1.00
Ratio of Students Requiring Remedial Science Course(s)	8388	0.62 (0.22)	0.00	1.00
Ratio of Students Involved in a Gifted & Talented Program	8388	0.06 (0.01)	0.00	1.00

Table 5 presents the descriptive statistics, aggregating across all schools, for variables describing average percent of students who are homeless, average percent of students requiring special education programs, and average percentage of students qualifying for free or reduced lunch at the per school level. These metrics were calculated after the data aggregation method took place to compile and validate the school level dataset.

Table 5: Descriptive Statistics for School Level Homeless, Special Education, and Free/Reduced Lunch Variables

Variable	N	Mean% (SD)	Min%	Max%
Percent of Homeless School Population	20	2.65 (3.01)	0.40	11.70
Percent of Students Requiring Special Education Programs	20	18.82 (2.3)	4.20	33.10
Percent of Students Qualifying for Free or Reduced Lunch	20	89.34 (9.69)	34.00	100.00

5.5 Procedure

Data analysis occurred in five unique phases. The first phase consisted of the data cleaning, data processing, and master data file aggregation that has been written about in detail above. During this phase, standard practices were taken to monitor the validity of the joins and aggregation.

The second phase consisted of exploratory descriptive analysis that was used to justify the use of, not only the dataset as a whole, but the individual grades and schools within this dataset. The results of the phase have been detailed in the previous section. The goals of this phase were to analyze the data, post-cleaning, to ensure the removal of certain components of dirty or missing data did not compromise the overall generalizability of the dataset. During this phase, it was also discovered that the data did not support analysis using a student's "academic career" as one observation. To utilize the data in such a way, all of the student's data from each school/grade would have to exist as one unique observation. The issues that arose from attempting to utilize the data in such a way came from how the data was collected and stored prior to data acquisition.

In table 6, descriptive statistics at the school level describing patterns of student turnover (transfer out of district), student transfer within district, and student dropout are presented. These variables either existed in the dataset as flags or were created from data representative of a student's arrival into or departure from a specific school code.

Table 6: Descriptive Statistics for School Level Turnover, Transfer within District, and Dropout Variables

Variable	N	Mean% (SD)	Min%	Max%
Percent of Student Turnover (Transfer Out of District)	20	28.48 (6.94)	8.10	61.00
Percent of Students that Transfer within District	20	49.87 (7.10)	6.40	71.00
Percent of Students that Dropout	20	4.40 (0.89)	0.00	17.30

After viewing the amount of data that would have to be removed due to significant portions missing at the student level, or inequalities at the school level, it was decided to analyze each grade level as a snapshot. This is beneficial for a number of reasons, primarily because it allows for the use of a more natural proportion of the data. The students that would be removed due to missing or inconsistent data patterns represented a large portion of the categorical outcome that did not attend higher education. By removing these students, it misrepresents the two samples and allows for the development of a model that is not generalizable on any other data. This would greatly hinder the purpose of this dissertation, due to the premise that supervised learning and data mining algorithms reliant on whole data sets are being utilized for model comparison against a portion of the data derived as a test/validation dataset.

A second reason this structure is beneficial for this dissertation is that it allows the comparison to be iterative, viewing each static grade level snapshot individually. From a model comparison approach, this allows for models to be compared on unique data at four different instances. When 9^{th}, 10^{th}, 11^{th}, and 12^{th} grade are analyzed individually, it reveals if any modeling techniques have

advantages when analyzing data with a greater portion of noise (considering academic performance is observed to be less polarizing at certain grade levels).

The third phase consisted of splitting the clean data into a train dataset and a test dataset. The purpose of splitting the master dataset into two validation sets is to fulfill the requirements for the supervised learning component of model comparison. The models were calibrated and weighted based on the input data contained in the train dataset. Then the calibrated models were used to analyze test dataset, treating the outcome variable as if it were unknown. Once the predicted outcome variable was collected, it was then compared to the known outcome variable allowing for the creation of the confusion matrix. The two datasets would be near exact in size (50% train / 50% test) to maintain the likelihood of proper school level sample sizes for the models requiring nested model structures. To accomplish this data manipulation, a Statistica data mining workspace was built for two subsets with the approximate split percentage set to 50. Once this was done, the datasets were imported into SAS for the multilevel logistic model and stored in an in-memory Statistica workspace for the gradient boosted trees and artificial neural networks.

The datasets would also be identical across all models; therefore, no resampling would be done between model development. The data splitting process only take place once per grade level. This helped to maintain the most appropriate comparison across models, allowing for improper sampling to be ruled out as a potential detractor. The process of splitting the data would take place at the grade level, so the final outcome contained a train dataset and a test dataset for 9^{th}, 10^{th}, 11^{th}, and 12^{th} grade.

The fourth phase consisted of model building. During this phase multilevel logistic regression models, gradient boosted decision trees, and artificial neural networks were created with the focus of predicting the enrollment behavior of a set of students. This was performed on each grade level included in high school, 9^{th} – 12^{th} grade, individually using the data that was cleaned, processed, and split during the previous three phases. The development of these three models will be detailed in subsections dedicated to each modeling type.

5.5.1 Gradient Boosted Decision Trees

Boosted Tree Models were developed for each grade level from 9^{th} grade through 12^{th} grade. These models, unlike the multilevel logistic regression models, are iterative during the model development, allowing the gradient component to continue recalibrating the model with prior information gained from the previous complete iteration. Due to this process, computation time can take longer, but less time is spent validating variable selection because that process is implicit in the model development. These models were developed in Visual Basic (SVB) using the Statistica Data Mining platform. Model outputs were stored as Predictive Model Markup Language (PMML), which is an XML based language that is used to store and exchange model information between multiple datasets. SVB was also used to analyze test dataset for each grade using the calibrated models PMML code derived from the training dataset. All fit measures and confusion matrices were created in Statistica.

In developing the gradient boosted decision tree for each grade, the model was created with a learning rate of 0.100. The learning rate of a gradient boosted

decision tree acts as an additive shrinkage parameter that is applied to the consecutive model estimates occurring with each additional iteration through the model. Friedman (1999) stated that learning rates of 0.100 and lower provide the best predictive accuracy. The most conservative value on the range suggested by Friedman was selected for this model. After the additive modifier (learning rate) is applied to the model, the boosting step occurs. During this step of development, the prediction residuals for an independently drawn sample of observations are computed and that information is used to better model the data during the next iteration (Blagus & Lusa, 2017; Mayr, Binder, Gefeller, & Schmid, 2014).

This gradient boosted decision model was allocated 200 additive terms selected for processing. This allows the algorithm to compute 200 simple decision trees using successive bootstrapping. Once these 200 successive trees are created and tested, the model is designed to create another 200 trees if it detects that the final tree in the sequence is the best fit. This allows the algorithm to ensure future iterations of the model won't provide better estimates. This value was chosen because it should provide most models with enough iterations to aptly understand the relationships between each variable in the model. More successions could always be added to the default, but the trade-off is that addition of strenuous computation power and time requirements. It is quite possible that the models best performing iteration won't occur at the end of the additive steps, but instead the model performance will produce higher error rates as it approaches the 200^{th} step. This is common practice in machine learning model, and helps the researcher identify that the model does not need any additional iterations.

The gradient boosted decision tree model was created with a minimum child node value of 1 and maximum child node value of 3. These means that at each split in each decision tree, the maximum number tree size will be one root node with three child nodes. These values were selected because a larger number of splits in the parent nodes leads to overfitting a model and losing generalizability (Hausman, Abrevaya, & Scott-Morton, 1998). Overfitting occurs when the model development is too precise and replicates the training data too closely. Without controlling for it with proper modeling practice, overfitting can occur anytime large bodies of variables are presented to a training model (Cawley & Talbot, 2007).

With each iteration, the training data is analyzed and the corresponding model is evaluated using the test data. The prediction results are utilized to calculate the average deviance at each iteration. This metric is useful in identifying the best fitting model throughout all iterations. The average multinomial deviance is comparable to the -2loglikelihood fit statistic, but when pertaining to decision trees, the saturated model assigns a probability of one to each observation, since the test dataset contains full information of the actual outcomes being used to determine fit. The function in use to determine this fit is:

$$-2\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\, log\left(\frac{\hat{p}_j(x_i)}{y_{ij}}\right)$$

Equation 13

Where $y_i$is a k-vector indicating which class observations *i* belong in the training dataset, and $\hat{p}_j(x)$ is a vector of probabilities estimated by the model.

During the tree calibration, the model is also iteratively measuring relative and global variable importance. This is done by measuring the number of times a

variable is used as the decision component in a node split. The function implemented in the gradient boosted decision trees in this study was:

$$\hat{\imath}_t^2(T) = \sum_{t=1}^{J-1} \hat{\imath}_t^2 1(v_t = j)$$

Equation 14

Where *t* represents the nonterminal nodes and *J* represents the terminal nodes in the terminal node tree, T. v_t is the splitting variable for the parent node (*t*), and $\hat{\imath}_t^2$ is the observed improvement across the iterations in squared error due to the parent node split (Friedman, 2001). This function allows for the measure of the overall importance a variable maintains when viewed with all other variables in the data. As stated above, the expected importance score is directly related to the number of times a variable was used to split the data, or 'make a decision'. This split was decided on by the algorithm because it improved the performance of the measure when weighted by all other nodes that fall under the split (Elith, Leathwick & Hastie, 2008).

The risk estimate produced from the model development represents the proportion of cases incorrectly classified from the sample of data adjusted by the unequal misclassification costs. This correction must exist because, due to the supervised learning procedure used for model validation, the empirical distribution is defined by the training set sampled from the full data prior to the analysis. The function in place to determine this estimate is:

$$\hat{R}(f) = \sum_{i=1}^{n} L(y_i, f(x_i))$$

Equation 15

This metric is output with the final model results after tree selection and variable selection has taken place.

5.5.2 Artificial Neural Networks

Artificial neural networks were developed for each grade level from grade 9 through grade 12. In the same way as decision trees, these models are developed iteratively through algorithmic functions. Due to this component of the model development, manual checks were not required for variable selection. The training data is processed during the neural network development, and the variables are ranked by importance, those deemed unimportant removed from the model.

Similar to the variable importance metric created for the gradient boosted decision trees, artificial neural networks utilize Global Sensitivity Analysis (GSA) to determine the optimal variables used in the network model. This function is similar to the variable importance calculation, except the scale for global sensitivity is reversed. Pianosi, Sarrazin & Wagener (2015) defined GSA as a set of mathematical procedures implemented to investigate how variance in model output can be credited to model inputs.

By design, neural networks create bundles of smaller relationships, with unique weights between nodes (input or hidden), calibrate the individual models iteratively with weighted paths, and test whether or not the presence of a variable increases or decreases error in estimation. During this process of node assignment and path weighting, reduction of variance tests are applied at every level to determine which variable should be utilized. The deviance of each node in any given model is defined as:

$$D_i = -2 \sum_k n_{ik} \ln(p_{ik})$$

Equation 16

where p_{ik} represents the probability distribution (probabilities are unknown at this time and will be estimated from the proportional classifications at the individual node), *i* represents the iteration in the process, *k* is the class, and n_{ik} is the total number of classes used for calculation (Breiman et al, 1998). With the above function representing deviance at a given node, the function for the reduction of deviance from parent node (*t*) into child nodes (*u* and *v*) is defined as:

$$D_t - D_u - D_v = 2 \sum_k \left[n_{uk} \ln \left(\frac{n_{uk} n_t}{n_{tk} n_u} \right) + n_{vk} \ln \left(\frac{n_{vk} n_t}{n_{tk} n_v} \right) \right]$$

Equation 17

The calculation of the GSA takes into account the node paths that were selected for the model, as well as, those that were tested and not accepted for the model (Venables & Ripley, 1997). The reduction of deviance for each node is summed over the entire network, then the variable sensitivity, or GSA, can be calculated as (Brieman et al., 1998):

$$M(x_m) = \sum_{t \in T} \Delta I \, (S_m, t)$$

Equation 18

Where *I* represents the deviance for a split, *t* represents a specific node, *T* represents the set of all nodes, and s_m is the competing node. The competing node was the node that was iteratively calculated during the calibration process, but not selected for the final model.

For the models developed during this study, a GSA threshold of 0.9 or greater was implemented for variable selection. This was decided on due to the steep drop off in GSA scores after the 0.9 value.

For inter-model comparison across grade level, the ROC Area will also be reported and calculated. Every model produces an Receiver Operating Characteristic (ROC) Curve (Fawcett, 2006). This curve represents a measure of both specificity and sensitivity (King, 2003). The ROC Area represents the area under the curve, with a higher area characterizing a better model (Hastie, Tibshirani, & Friedman, 2009).

These models were developed in SVB using the Statistica Data Mining platform. Model outputs were stored as PMML, so they could be utilized on the test dataset. All fit measures and confusion matrices were created in Statistica. Due to excessive length, the PMML code output was not appended to the study.

5.5.3 Multilevel Logistic Regression

One multilevel logistic regression model was created for each grade level from 9^{th} grade through 12^{th} grade. These models were all created with the college enrollment as the predicted outcome. During the modeling process, the SAS procedures PROC QUANTSELECT, PROC GLMSELECT, and PROC PLM were utilized to validate the variable selection in the training data. PROC QUANTSELECT was used to split the data into two similar datasets, PROC GLMSELECT was used to automate the variable selection process, and PROC PLM was used to score the test data set using the model developed from the training dataset, allowing for the creation of a confusion matrix. This process allowed the

comparison of fit across all models, since the two data mining models do not support a p-value by design.

PROC GLMSELECT was utilized so that the variable selection methodology would always reside on the machine side for each model. To implement this procedure, a GLMSELECT was deployed on the data using a ridge regression decision function, as well as a second implementation on the same data using LASSO regression (Tibshirani, 1996). This was done because LASSO models produce less consistent results if there are issues with collinearity (Byon, Shrivastava, & Ding, 2010). The results of both models were compared to diagnose differences and choose the best variable pool.

PROC GLIMMIX, was utilized to build the models with student level variables at level-1 and aggregated school level variables at level-2. The STORE function of PROC GLIMMIX was utilized as the training data was analyzed, so that the model could be recalled and the weights could be applied to the test data. As stated previously, PROC PLM was used to call the stored scoring model from PROC GLIMMIX and calculate the confusion matrix.

5.5.4 Model Comparison

The final phase of the analysis was the model comparison phase. During this phase, fit statistics and confusion matrix outputs were compiled for all grades and models. The interest was in, not only, how each compared across one specific grade, but also how an individual model type faired at predicting similar outcomes at every grade.

5.6 Summary

This chapter provided a summary of the sample descriptive statistics and methodology that went into data processing, model development, and model comparison to determine which model best estimates college enrollment using high school data. A description of the methodology behind each model type, multilevel logistic regression, gradient boosted decision trees, and artificial neural networks, created a foundation for understanding how each model was implemented. Data backed justification was provided as to why the grade levels would be analyzed independently, and additional interest was expressed in individual model performance as grade level changes. After the modeling summary, an explanation of the model comparison outlined how the models would compared. The following chapter presents the results from the models outlined above.

CHAPTER VI: RESULTS

6.1 Overview

This chapter presents the results of the multilevel logistic regression, gradient boosted decision trees, and artificial neural networks, at each grade level, 9th, 10th, 11th, and 12th. The goal of this model comparison is to answer the question presented in chapter three; which model will most adequately predict college enrollment using data collected throughout the students' time in high school. A secondary interest in this study is not only which model performs the best under a supervised learning environment, but also which performs the most consistently across independently evaluated grade levels.

6.2 Gradient Boosted Decision Tree Model Results

6.2.1 Gradient Boosted Decision Trees – Grade 9

The optimal gradient boosted decision tree for the 9th grade dataset was located in the 195th additive tree created. The average error rate of correctly classified cases from the model based on comparison of training and test was decreased to 0.2359 with this tree. The selection rate was set to 200, which caps the number of trees created to 200.

After the model was optimized on the training data, the predictor importance algorithm selected 14 variables to be utilized. This selection was done by the variable importance calculations explained above. Table 7 below presents the variables used for the 9th grade model along with the corresponding Predictor Importance score. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 1: Grade 9 Gradient Boosted Decision Tree Predictor Importance

Variable	Predictor Importance Score
STEMCourseGPA	1.000
TotalDaysOnRoll	0.953
NonSTEMCourseGPA	0.828
AttendanceRate	0.786
AboveGradeLevelMathCourse	0.431
RemedialScience	0.326
AboveGradeLevelScienceCourse	0.283
RemedialMath	0.276
TotalDaysSuspended	0.123
EOIBiologyScore	0.098
ACTComp	0.091
TotalReferrals	0.086
EOIAlgebraIScore	0.067

The overall model risk estimate was 0.2471, which is the inverse of the model accuracy calculated with the confusion matrix. Table 9 below presents the primary model fit statistics being used for the overall comparison. This table contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics. For the introduction to how the fit measures are calculated, and what they represent in the model, please reference Chapter Three.

Table 9: Grade 9 Gradient Boosted Decision Tree Model Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8104
Specificity	0.7491
Precision	0.5107
Accuracy	0.7641
MCC	0.4934

6.2.2 Gradient Boosted Decision Trees – Grade 10

The optimal gradient boosted decision tree for the grade 10 dataset was located in the 188th additive tree created. The average error rate of correctly

classified cases from the model based on comparison of training and test was decreased to 0.2146 with this tree, which was slightly better than the grade 9 model.

After the model was optimized on the training data, the predictor importance algorithm selected 12 variables to be utilized. This selection was done by the variable importance calculations explained above. The number of variables utilized was likely smaller than the grade 9 model, due to a lessened number of standardized tests that took place in grade 10. The standardized testing variables, representing individual EOI exams, utilized in the Grade 9 model were not highly rated on the predictor importance output, so the lack of standardized test data for certain grades is not a concern. Table 10 below presents the variables used for the 10th grade model along with the corresponding Predictor Importance score. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 10: Grade 10 Gradient Boosted Decision Tree Predictor Importance

Variable	Predictor Importance Score
TotalDaysOnRoll	1.000
STEMCourseGPA	0.971
NonSTEMCourseGPA	0.768
AttendanceRate	0.647
AboveGradeLevelMathCourse	0.397
AboveGradeLevelScienceCourse	0.377
ACTComp	0.238
RemedialScience	0.171
RemedialMath	0.158
TotalReferrals	0.075
TotalDaysSuspended	0.069
TotalAPCoursesTaken	0.062

The overall model risk estimate was 0.2372, which shows a minor improvement over the grade 9 model. Table 11 contains the Sensitivity, Specificity,

Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 10 Gradient Boosted Decision Tree Model.

Table 11: Grade 10 Gradient Boosted Decision Tree Model Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8092
Specificity	0.7770
Precision	0.5602
Accuracy	0.7854
MCC	0.5310

6.2.3 Gradient Boosted Decision Trees – Grade 11

The optimal gradient boosted decision tree for the grade 11 dataset was located in the 573rd additive tree created. This model displayed the final additive iteration to be the most accurate predictor, thus triggering the addition of 200 more trees to further optimize the model. This occurred twice for a total of 600 additive trees. The process took longer computationally, but significantly increased the prediction precision by adding the additional trees. The average error rate of correctly classified cases from the model based on comparison of training and test was decreased to 0.2016 with this tree, once again decreasing error when compared to the previous grade level.

After the model was optimized on the training data, the predictor importance algorithm selected 13 variables to be utilized. This selection was done by the variable importance calculations explained above. Similar to the grade 10 model, the presence of EOI variables were not as frequent in the data, but ACT scores became much more common in the Grade 11 data. The number of variables utilized was very close to the grade 10 model, but still smaller than the grade 9 model.

Table 12 below presents the variables used for the 11th grade model along with the corresponding Predictor Importance score.

Table 12: Grade 11 Gradient Boosted Decision Tree Predictor Importance

Variable	Predictor Importance Score
TotalDaysOnRoll	1.000
STEMCourseGPA	0.871
AttendanceRate	0.845
AboveGradeLevelMathCourse	0.662
AboveGradeLevelScienceCourse	0.589
NonSTEMCourseGPA	0.421
ACTComp	0.299
RemedialMath	0.296
RemedialScience	0.247
TotalReferrals	0.224
EOIAlgebraIIScore	0.197
TotalDaysSuspended	0.167
TotalAPCoursesTaken	0.082

The overall model risk estimate was 0.2053, which shows a minor improvement over the grade 10 model. Table 13 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 11 Gradient Boosted Decision Tree Model. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 13: Grade 11 Gradient Boosted Decision Tree Model Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8210
Specificity	0.7830
Precision	0.6212
Accuracy	0.7945
MCC	0.5664

6.2.4 Gradient Boosted Decision Trees – Grade 12

The optimal gradient boosted decision tree for the grade 12 dataset was located in the 683rd additive tree created. This model displayed the final additive

iteration to be the most accurate predictor, thus triggering the addition of 200 more trees to further optimize the model. This occurred three time for a total of 800 additive trees. Similar to the grade 11 model, the process took longer computationally, but yielded better results. The increase in required trees resulted in a decrease in average error rate of correctly classified cases from the model based on comparison of training and test to a value of 0.1522

After the model was optimized on the training data, the predictor importance algorithm selected 13 variables to be utilized. This selection was done by the variable importance calculations explained above. Similar to the grade 11 model, the presence of EOI and EXPLORE variables were not adequately distributed within the data, but ACT scores became much more common in the Grade 12 data. The number of variables utilized were the same as the grade 11 model, but, once again, smaller than the grade 9 model. Table 14 below presents the variables used for the 12th grade model along with the corresponding Predictor Importance score.

Table 14: Grade 12 Gradient Boosted Decision Tree Predictor Importance

Variable	Predictor Importance Score
STEMCourseGPA	1.000
TotalDaysOnRoll	0.839
AttendanceRate	0.777
NonSTEMCourseGPA	0.651
ACTComp	0.637
AboveGradeLevelScienceCourse	0.525
TotalAPCoursesTaken	0.391
AboveGradeLevelMathCourse	0.378
RemedialMath	0.325
DaysSuspended	0.229
TotalReferrals	0.158
EOIAlgebraIIScore	0.151
RemedialScience	0.150

The overall model risk estimate was 0.1823, which is slightly smaller than the grade 11 model. Table 15 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 12 Gradient Boosted Decision Tree Model. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 15: Grade 12 Gradient Boosted Decision Tree Model Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8475
Specificity	0.8480
Precision	0.7414
Accuracy	0.8478
MCC	0.6759

6.3 Artificial Neural Networks Model Results

6.3.1 Artificial Neural Networks – Grade 9

The artificial neural network developed for Grade 9 data contained an ROC Area value of 0.8688. The program in use was written to develop 200 artificial neural networks, select the five best models, and resample to test and identify the best remaining model. The primary model selected was then used to analyze the test data set in the supervised learning environment, similar to the other two models in the study. The multi-layer perceptron (MLP) network developed using a softmax activation function and the error function set to cross-entropy. Cross entropy was selected in the training phase of the model due to it's enhanced performance with classification outcomes in neural networks (Bishop, 1995). Weight decay with a maximum value of 0.01 and minimum value of 0.001 was applied to the hidden layer nodes created during calibration of the model. The application of weight decay

to the hidden layer nodes modifies the model's error function to penalize larger weights. This is implemented to maintain smaller weights and reduce the chances of overfitting the model (Byon, Shrivastava, & Ding, 2010). The activation function was set to softmax by default because of the restraint applied when selecting cross entropy as the error function. As mentioned previously, MLP was selected as the calibration framework to better enhance the use of back-propagation.

Similar to the Grade 9 gradient boosted decision tree model, the inclusion of more standardized testing data in the form of EOI exams, allowed for easier use in the model. The GSA selected 10 variables to be used for the model, which is significantly less than the gradient boosted decision tree for the grade 9 model. Table 16 below presents the variables used for the grade 9 model along with the corresponding GSA values. The weights assigned to node relationships and hidden nodes created for the final model are presented in Appendix

Table 16: Grade 9 Artificial Neural Network Global Sensitivity Analysis

Variable	Global Sensitivity Analysis Score
STEMCourseGPA	0.939
NonSTEMCourseGPA	0.995
TotalDaysOnRoll	1.023
AttendanceRate	1.024
AboveGradeLevelMathCourse	1.047
TotalReferrals	1.071
DaysSuspended	1.107
EOIAlgebraIScore	1.320
ACTComp	1.364
AboveGradeLevelScienceCourse	2.941

Table 17 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 9 Artificial Neural

Network Model. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 17: Grade 9 Artificial Neural Network Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.7707
Specificity	0.7712
Precision	0.8180
Accuracy	0.7709
MCC	0.5379

6.3.2 Artificial Neural Networks – Grade 10

The artificial neural network developed for the Grade 10 data had an ROC area of 0.8784, slightly better than that of the Grade 9 artificial neural network model. The program in use was written to develop 200 artificial neural networks, select the five best models, and resample to test and identify the best remaining model. The primary model selected was then used to analyze the test data set in the supervised learning environment, analogous to the other two models in the study. Similar to the grade 9 model, the MLP network was developed using a softmax activation function and the error function was set to cross-entropy. Weight decay with a maximum value of 0.01 and minimum value of 0.001 was applied to the hidden layer nodes created during calibration of the model. The GSA variable selection kept 14 variables, which is more than existed in the grade 9 model. Table 18 below presents the variables used for the grade 10 model along with the corresponding GSA values.

Table 18: Grade 10 Artificial Neural Network Global Sensitivity Analysis

Variable	Global Sensitivity Analysis Score
STEMCourseGPA	0.996
TotalDaysOnRoll	0.998
NonSTEMCourseGPA	1.020
DaysSuspended	1.033
AboveGradeLevelMathCourse	1.041
TotalReferrals	1.047
AttendanceRate	1.064
RemedialMath	1.125
EOIAlgebraIScore	1.189
EOIReadingLA2Score	2.885
TotalAPCoursesTaken	2.899
AboveGradeLevelMathCourse	2.902
ACTComp	3.003
RemedialScience	3.981

Table 19 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 10 Artificial Neural Network Model. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 19: Grade 10 Artificial Neural Network Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8182
Specificity	0.7791
Precision	0.8182
Accuracy	0.8005
MCC	0.5973

6.3.3 Artificial Neural Networks – Grade 11

The artificial neural network developed for the Grade 11 data had an ROC area of 0.887, which is still an increase over the grade 10 ROC area value, but not quite as large of a jump as experienced from the Grade 9 model. The program in use was written to develop 200 artificial neural networks, select the five best models, and resample to test and identify the best remaining model. The primary model

selected was then used to analyze the test data set in the supervised learning environment, analogous to the other two models in the study. Similar to the other artificial neural network models developed, the MLP network was implemented using a softmax activation function and the error function was set to cross-entropy. Weight decay with a maximum value of 0.01 and minimum value of 0.001 was applied to the hidden layer nodes created during calibration of the model. The GSA variable selection kept 13 variables in the model. Table 20 below presents the variables used for the grade 11 model along with the corresponding GSA values.

Table 20: Grade 11 Artificial Neural Network Global Sensitivity Analysis

Variable	Global Sensitivity Analysis Score
TotalDaysOnRoll	0.992
STEMCourseGPA	0.996
TotalReferrals	1.001
NonSTEMCourseGPA	1.002
DaysSuspended	1.009
AttendanceRate	1.012
AboveGradeLevelMathCourse	1.035
TotalAPCoursesTaken	1.047
ACTComp	1.126
RemedialMath	1.200
AboveGradeLevelScienceCourse	1.321
RemedialScience	1.878
EOIAlgebraIIScore	1.975

Table 21 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 11 Artificial Neural Network Model. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 21: Grade 11 Artificial Neural Network Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8255
Specificity	0.7927
Precision	0.7834
Accuracy	0.8083
MCC	0.6175

6.3.4 Artificial Neural Networks – Grade 12

The artificial neural network developed for the Grade 12 data had an ROC area of 0.8926, which is still maintaining the trend of increasing as grade level increases. The program in use was written to develop 200 artificial neural networks, select the five best models, and resample to test and identify the best remaining model. The primary model selected was then used to analyze the test data set in the supervised learning environment, analogous to the other two models in the study. Similar to the other artificial neural network models developed, the MLP network was implemented using a softmax activation function and the error function was set to cross-entropy. Weight decay with a maximum value of 0.01 and minimum value of 0.001 was applied to the hidden layer nodes created during calibration of the model. The GSA variable selection kept 13 variables in the model. Table 22 below presents the variables used for the grade 12 model along with the corresponding GSA values.

Table 22: Grade 12 Artificial Neural Network Global Sensitivity Analysis

Variable	Global Sensitivity Analysis Score
STEMCourseGPA	0.993
AttendanceRate	0.997
NonSTEMCourseGPA	1.001
TotalDaysOnRoll	1.009
TotalReferrals	1.012
ACTComp	1.044
RemedialMath	1.090
DaysSuspended	1.101
AboveGradeLevelMathCourse	1.167
TotalAPCoursesTaken	1.209
RemedialScience	1.231
AboveGradeLevelScienceCourse	1.806
EOIAlgebraIIScore	1.875

Table 23 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 12 Artificial Neural Network Model. For a list of variables and corresponding descriptions, reference Table 8 in the appendix.

Table 23: Grade 12 Artificial Neural Network Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8293
Specificity	0.8077
Precision	0.8095
Accuracy	0.8184
MCC	0.6370

6.4 Multilevel Logistic Regression Model Results

6.4.1 Multilevel Logistic Regression Models – Grade 9

The model output from PROC GLMSELECT revealed that the optimal variables for the grade 9 model were STEMCourseGPA, AttendanceRate, NonSTEMCourseGPA, TotalReferrals, TotalDaysonRoll, and DaysSuspended. There were fewer variables identified when compared to the two data mining

models. This is most likely due to the pre-processing that takes when the networks are developed. The package used for analysis approximates non-random missing data using Gausian Basis Function Networks (Tresp, Ahmad, & Neuneier, 1994). The grade 9 multilevel logistic model can be viewed as:

$$Y_{ij} = B_{00} + B_{1j}(STEMCourseGPA)_{ij} + B_{2j}(AttendanceRate)_{ij} + B_{3j}(NonSTEMCourseGPA)_{ij} + B_{4j}(TotalReferrals)_{ij} + B_{5j}(TotalDaysOnRoll)_{ij} + B_{6j}(DaysSuspended)_{ij} + r_{ij}$$

where

$$B_{00} = \Upsilon_{00} + \alpha_{0j} + u_{0j}$$

Equation 19

Υ_{00} is the model grand mean, α_{0j} represents the effect unique to the school the student comes from the corresponding variable, and u_{0j} is the error associated with any predictions made at level-2 of the model. Results from the grade 10 model are located in Table 24.

Table 24: Grade 9 Multilevel Logistic Regression Results

Effect	Estimate	Standard Error	Df	t value	pr |t|
Intercept	-7.1124	0.1267	3027	-6.57	< .0001
STEMCourseGPA	0.6815	0.0028	13977	7.57	< .0001
AttendanceRate	2.5734	0.1924	13977	8.80	< .0001
TotalReferrals	-0.0414	0.0164	13977	-3.28	< .001
NonStemCourseGPA	0.2044	0.0265	13977	2.31	< .0001
TotalDaysOnRoll	0.0185	0.0008	13977	5.39	< .0001
DaysSuspended	-0.0001	0.0049	13977	-0.03	NS

Fit statistics reported -2 Res Log-Likelihood = 13852.69, AIC = 13856.69, and BIC = 13859.48. Table 25 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 9 multilevel logistic regression model.

Table 25: Grade 9 Multilevel Logistic Regression Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.7769
Specificity	0.7745
Precision	0.7426
Accuracy	0.7603
MCC	0.5214

6.4.2 Multilevel Logistic Regression Models – Grade 10

The model output from PROC GLMSELECT revealed that the optimal variables for the grade 10 model were STEMCourseGPA, AttendanceRate, NonSTEMCourseGPA, TotalReferrals, TotalDaysonRoll, and DaysSuspended. As with the grade 9 multilevel model, there are fewer variables included in the model than the comparable data mining models. The grade 10 multilevel logistic model can be viewed as:

$$Y_{ij} = B_{00} + B_{1j}(STEMCourseGPA)_{ij} + B_{2j}(AttendanceRate)_{ij} + B_{3j}(NonSTEMCourseGPA)_{ij} + B_{4j}(TotalReferrals)_{ij} + B_{5j}(TotalDaysOnRoll)_{ij} + B_{6j}(DaysSuspended)_{ij} + r_{ij}$$

where

$$B_{00} = \Upsilon_{00} + \alpha_{0j} + u_{0j}$$

Equation 20

Υ_{00} is the model grand mean, α_{0j} represents the effect unique to the school the student comes from the corresponding variable, and u_{0j} is the error associated with any predictions made at level-2 of the model. Results from the grade 10 model are located in Table 26.

Table 26: Grade 10 Multilevel Logistic Regression Results

Effect	Estimate	Standard Error	Df	t value	pr |t|
Intercept	-7.4988	0.0364	2158	-9.47	< .0001
STEMCourseGPA	0.5632	0.0036	5843	5.73	< .0001
AttendanceRate	1.5781	0.0061	5843	7.01	< .0001
TotalReferrals	-0.1126	0.0958	5843	-2.11	< .001
NonStemCourseGPA	0.0954	0.1390	5843	3.99	< .0001
TotalDaysOnRoll	0.0236	0.0949	5843	3.81	< .0001
DaysSuspended	-0.0017	0.9016	5843	-0.355	NS

Fit statistics reported -2 Res Log-Likelihood = 13993.17, AIC = 13996.17, and BIC = 13997.80. Table 27 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 10 multilevel logistic regression model.

Table 27: Grade 10 Multilevel Logistic Regression Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.7972
Specificity	0.7971
Precision	0.8028
Accuracy	0.7972
MCC	0.5942

6.4.3 Multilevel Logistic Regression Models – Grade 11

The model output from PROC GLMSELECT revealed that the optimal variables for the grade 11 model were STEMCourseGPA, AttendanceRate, TotalReferrals, NonSTEMCourseGPA, TotalDaysonRoll, RemedialMath, and ACTComp. As with the previous multilevel models, there are fewer variables included in the model than the comparable data mining models. The grade 11 multilevel logistic model can be viewed as:

$$Y_{ij} = B_{00} + B_{1j}(STEMCourseGPA)_{ij} + B_{2j}(AttendanceRate)_{ij} +$$

$$B_{3j}(TotalReferrals)_{ij} + B_{4j}(NonSTEMCourseGPA)_{ij} +$$

$$B_{5j}(TotalDaysOnRoll)_{ij} + B_{6j}(RemedialMath)_{ij} + B_{7j}(ACTComp)_{ij} + r_{ij}$$

where $$B_{00} = \Upsilon_{00} + \alpha_{0j} + u_{0j}$$

Equation 21

Υ_{00} is the model grand mean, α_{0j} represents the effect unique to the school the student comes from the corresponding variable, and u_{0j} is the error associated with any predictions made at level-2 of the model. Results from the grade 11 model are located in Table 28.

Table 28: Grade 11 Multilevel Logistic Regression Results

Effect	Estimate	Standard Error	Df	t value	pr |t|
Intercept	-8.6669	0.1780	1539	-6.10	< .001
STEMCourseGPA	0.5632	0.0034	5002	4.96	< .0001
AttendanceRate	3.7289	0.0015	5002	8.94	< .0001
TotalReferrals	-0.0291	0.0164	5002	-1.30	< .001
NonStemCourseGPA	-0.0439	1.0516	5002	-0.26	NS
TotalDaysOnRoll	0.0122	0.0019	5002	9.81	< .0001
RemedialMath	-0.0963	0.0383	5002	-3.55	< .001
ACTComp	0.2845	0.0101	5002	3.98	< .001

Fit statistics reported -2 Res Log-Likelihood = 13901.10, AIC = 13905.10, and BIC = 13907.40. Table 29 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 11 multilevel logistic regression model.

Table 29: Grade 11 Multilevel Logistic Regression Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.7795
Specificity	0.8210
Precision	0.7830
Accuracy	0.8022
MCC	0.6008

6.4.4 Multilevel Logistic Regression Models – Grade 12

The grade 12 data was analyzed using PROC GLMSELECT to determine the variables most suited for the grade level multilevel logistic regression model. The model output revealed that STEMCourseGPA, AttendanceRate, TotalReferrals, NonSTEMCourseGPA, TotalAPCoursesTaken, ACTComp, and TotalDaysOnRoll were the most suited variables to use for the GLIMMIX model development. The grade 12 model can be viewed as:

$$Y_{ij} = B_{00} + B_{1j}(STEMCourseGPA)_{ij} + B_{2j}(AttendanceRate)_{ij} + B_{3j}(TotalReferrals) + B_{4j}(NonSTEMCourseGPA)_{ij} + B_{5j}(TotalAPCoursesTaken)_{ij} + B_{6j}(ACTComp)_{ij} + Y_{07}(TotalDaysOnRoll) + r_{ij}$$

where

$$B_{00} = Y_{00} + \alpha_{0j} + u_{0j}$$

Equation 22

Y_{00} is the model grand mean, α_{0j} represents the effect unique the school has on the student data, and u_{0j} is the error associated with any predictions made at level-2 of the model. Results from the grade 12 model are located in Table 30.

Table 30: Grade 12 Multilevel Logistic Regression Results

Effect	Estimate	Standard Error	Df	t value	pr |t|
Intercept	-5.9353	0.1672	2139	-11.34	< .0001
STEMCourseGPA	0.5812	0.0049	4041	3.63	< .0001
AttendanceRate	1.8379	0.0071	4041	3.01	< .0001
TotalReferrals	-0.1195	0.0326	4041	-3.85	< .001
NonStemCourseGPA	0.2351	0.0959	4041	6.69	< .001
TotalAPCoursesTaken	0.0599	0.7679	4041	0.81	NS
ACTComp	0.2447	0.0094	4041	3.17	< .0001
TotalDaysOnRoll	0.0172	0.0499	4041	2.32	< .001

Fit statistics reported -2 Res Log-Likelihood = 13643.80, AIC = 13645.80, and BIC = 13646.74. Table 31 contains the Sensitivity, Specificity, Precision, Accuracy, and Matthew's Correlation Coefficient (MCC) metrics for the grade 12 multilevel logistic regression model.

Table 31: Grade 12 Multilevel Logistic Regression Performance Metrics

Fit Measure	Model Performance Score
Sensitivity	0.8249
Specificity	0.8153
Precision	0.8085
Accuracy	0.8200
MCC	0.6400

6.5 Model Comparison

By design, this study had two areas of focus, one being which model performed better at predicting college enrollment, and the second, which model was most successful at consistently estimating correct outcomes across grade level data. The across grade interest was, in part, due to the differences in estimation method, data processing, and implicit missing data correction when viewing the three models. The following sections will present a comparative analysis of the fit statistics produced by the models in the supervised learning scenario across grade within model and across model within grade.

The model fit statistics produced by the confusion matrix were used to create a grade level comparison across each model. Below, in Table 32, you will find the results from all models and grades.

Table 32: Grade & Model Level Performance Metrics

Variable	Model	Sensitivity	Specificity	Precision	Accuracy	MCC
Grade 9	GBDT	0.8104	0.7491	0.5107	0.7641	0.4934
	ANN	0.7707	0.7712	0.8180	0.7709	0.5379
	MLR	0.7769	0.7445	0.7426	0.7603	0.5214
Grade 10	GBDT	0.8092	0.7777	0.5602	0.7854	0.5310
	ANN	0.8182	0.7791	0.8182	0.8005	0.5973
	MLR	0.7945	0.7971	0.8056	0.7958	0.5914
Grade 11	GBDT	0.7907	0.8019	0.7577	0.7970	0.5903
	ANN	0.8255	0.7927	0.7834	0.8083	0.6175
	MLR	0.7795	0.8210	0.7830	0.8022	0.6008
Grade 12	GBDT	0.8475	0.8480	0.7414	0.8478	0.6750
	ANN	0.8293	0.8077	0.8095	0.8184	0.6370
	MLR	0.8249	0.8153	0.8085	0.8200	0.6400

GBDT represents the gradient boosted decision tree models, ANN represents the artificial neural network models, and MLR represents the multilevel logistic regression models. Accuracy and MCC are the most important performance metrics for model comparison in this study. As mentioned earlier, the Sensitivity metric, also referred to as the True Positive Rate, represents how often the prediction of an event happening is correct out of all predictions that the event happened. The Specificity metric represents the False Positive Rate, or how often the prediction of an event not happening is mistakenly predicted as the event happening. The Precision metric represents how often a correct even prediction occurs out of all instances the model says an event occurred. The Accuracy metric represents how often the classifier predicted correctly across all classifications. Lastly, the MCC (Matthew's Correlation Coefficient), or mean square contingency coefficient, exists on a -1 to 1 scale. This metric does the best job of representing the entire confusion matrix, and how well the overall classification model is doing. This stability is due

to the MCC's ability to control for unbalanced cell sizes in the matrix (Lin & Chen, 2012; Brodley & Friedl, 1999). A value of -1 represents a completely wrong classification model, while a 1 represents a perfect classification model.

6.5.1 Detailed Model Results

At the grade 9 level the GBDT performed the poorest overall (MCC = 0.4934), while the ANN (MCC = 0.5379) and MLR (MCC = 0.5214) experienced a similar level of classification success. The Accuracy metrics were all very close in grade 9. The most glaring deficit was the Precision score of 0.5107 experienced by the GBDT model. This is experienced when the model has successful predictions but predicts more non-occurrences correctly than actual occurrences.

At the grade 10 level, the same relationship was evident with ANN (MCC = 0.5973) and MLR (MCC = 0.5914), and GBDT performing significantly worse (MCC = 0.5310). Similar to the grade 9 models, the Accuracy metrics indicated ANN (0.8005) and MLR (0.7958) were the most accurate models, with ANN being the highest level of overall classification accuracy. Once again, the GBDT model experienced a very low Precision score of 0.5602 when compared to the other two models.

The grade 11 models showed the three models getting much closer in performance. The ANN model (MCC = 0.6175) had the best overall classification score, but the MLR model (MCC = 0.6008) and GBDT model (MCC = 0.5310) were very close. The separation between the models became even less apparent in the Accuracy scores, with the range between the worst and best models being

0.0113. The GBDT model Precision score (0.7577) is still less than the other two models, but now by the same margin.

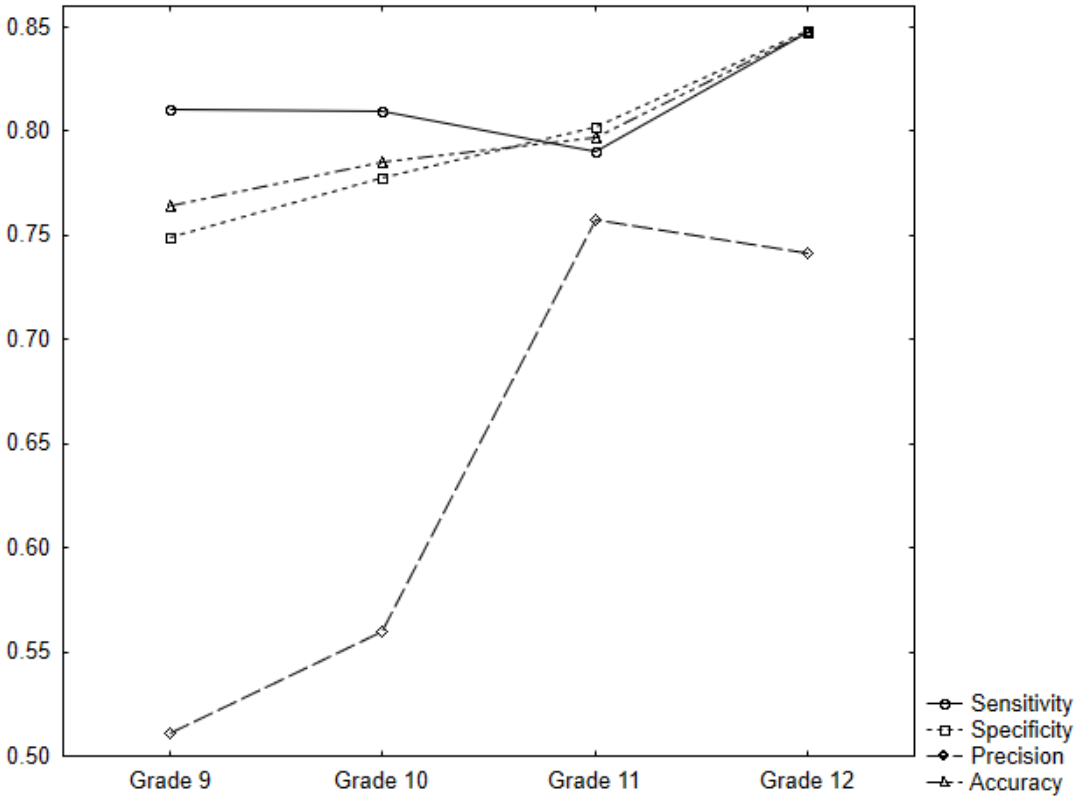
The grade 12 model results were quite different when compared to the other three grades. The ANN (MCC = 0.6370) and MLR (MCC = 0.6400) models were still very close in performance, but the GBDT (MCC = 0.6750) model outperformed both on almost every category. The model accuracy of the GBDT model was especially high at 0.8478. The only category the GBDT model was not the highest in was the Precision score (0.7414).

The ANN and MLR models maintained consistency in prediction success across all grades, with the ANN slightly higher than the MLR. The GBDT model performed at a less significant level by overpredicting non-occurrence events for the first three grades. Although, it was the worst performing model at the first three grade levels, the grade 12 data showed the GBDT model out-predicting both other models in predictive accuracy and the overall classification model by a large margin. This shows that the GBDT is more susceptible to overfitting and hurting generalizability when being applied to a test data set, but also showing significantly higher success when fitting the data appropriately.

The grade 12 data was the most representative of a student's profile before enrolling in higher education. There were also more variables for the model to choose from. The results show that the GBDT model is the best predictor of college enrollment based on the grade 12 data. It can be assumed that since the model relies heavily on node selection for splitting the data, the difference in unique academic variables from the grade 9 data to the grade 12 data caused improper node selection

for the split earlier in the trees. The analysis also supports the claim that if there are not checks in place to lessen the likelihood of overfitting, an ANN or MLR model might be more suitable.

Figure 2: GBDT Performance Metrics by Grade Level



As can be seen in Figure 2, the GBDT experienced noticeable inconsistencies with the Precision metric. The overall growth of the other three performance metrics increased, with Accuracy and Specificity increasing approximately 0.10. The MCC was withheld from Figure 2 due to the measurement scale being -1 to 1, rather than 0 to 1, like to the rest of the performance metrics.

Figure 3: ANN Performance Metrics by Grade Level

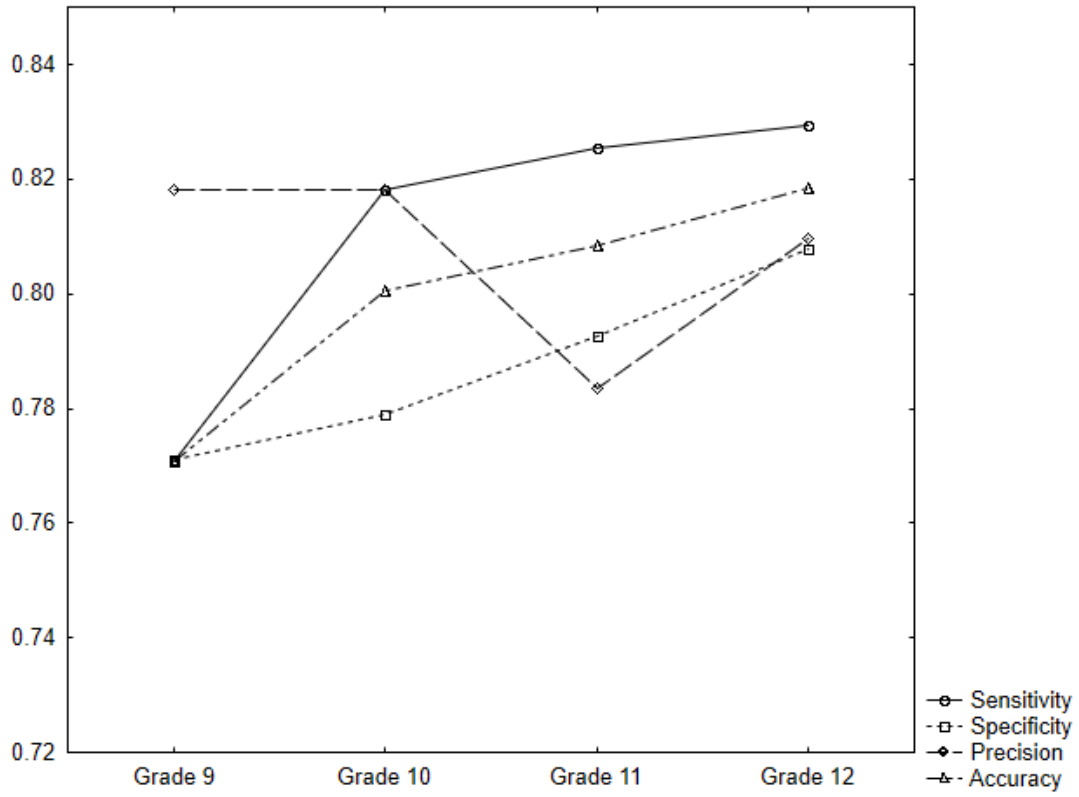


Figure 3 above displays the performance metrics across grade level for the ANN model. Once again, MCC was withheld from this figure. The ANN model, did not achieve high scores comparable with the GBDT model, but stayed much more consistent across all grade levels. Similar to Figure 2, the Precision metric was more erratic than the other metrics. Sensitivity, Specificity, and Accuracy all increased consistently from approximately 0.77 to approximately 0.83, a margin of only 0.5.

Figure 4: MLR Performance Metrics by Grade Level

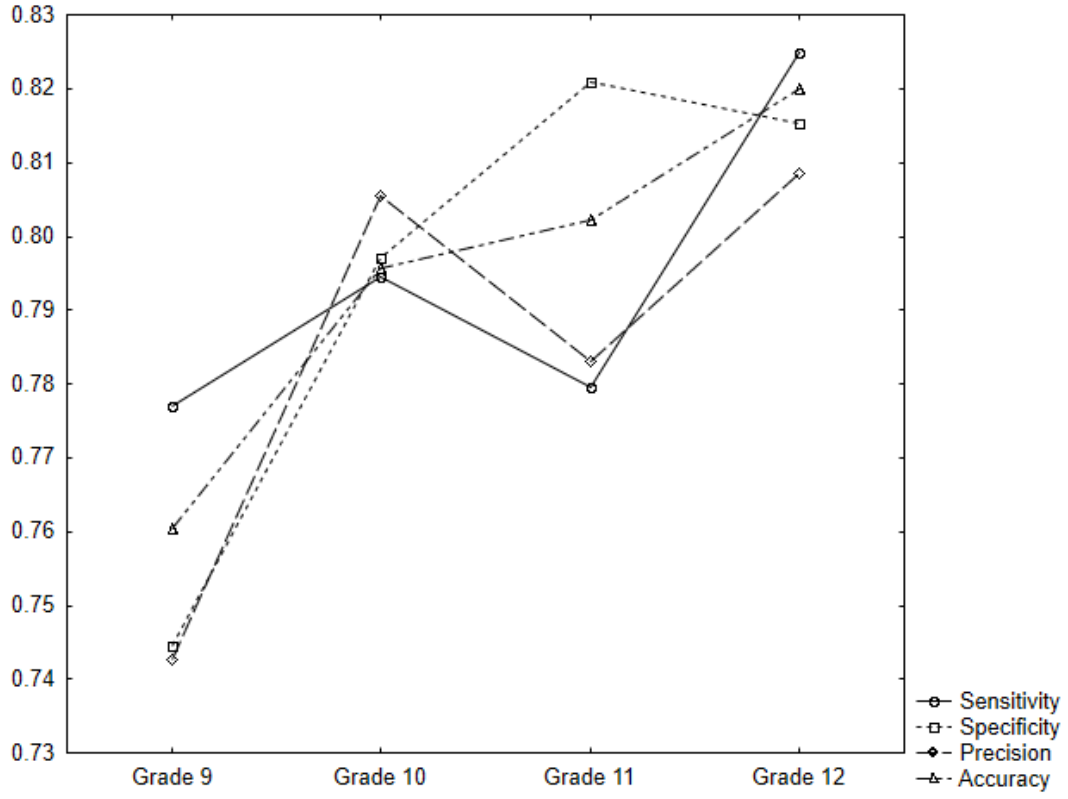


Figure 4 reveals that the MLR models conveyed much more consistency in growth. This is especially noticeable with the Precision metric that was much less predictable with the other models. This indicates that MLR models are less likely to overclassify as a non-event when modeling with data similar to the data used in this study. On average, the MLR models had lower Grade 9 scores than the ANN models, and they did not quite reach the ceiling that the ANN models achieved. This indicates that although the MLR models are less likely to overclassify a non-event, they are also less consistent with prediction across grade level.

Figure 5: MCC Performance Scores Across Grade and Model

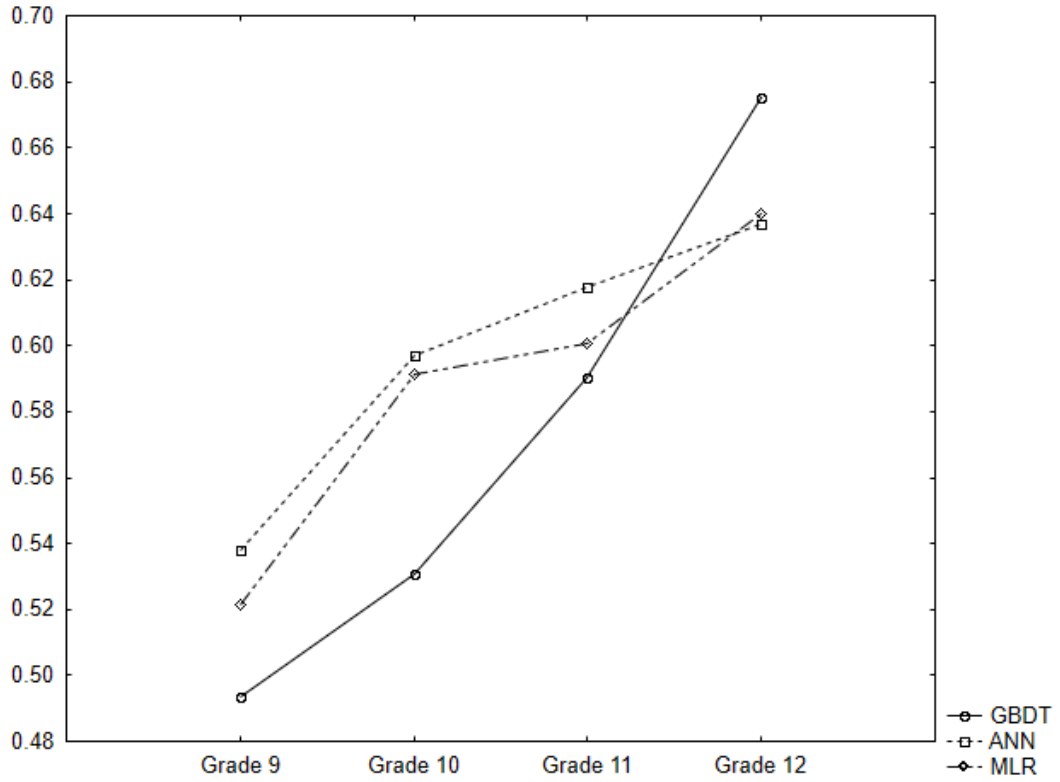


Figure 5 displays the MCC performance scores for all three models across all grade levels. This metric was the primary indicator of a successful classification model. Although the GBDT model performed poorer than the other models at the grade 9 level, it outgained both models very steadily as grade level progressed. It is also evident that the Grade 9 models were inferior across all models types.

6.5.2 Summary

The current chapter has provided results from multiple grade level models across three different analysis types, with comparable predictive accuracy measures in the style of supervised learning. This style of model comparison is typically not performed on traditional statistical measures, but due to lack of comparable

estimation method between the models, this was the most accurate way to show the efficacy of the classification models.

The chapter started with four unique grade level analyses using gradient boosted decision tree models. This was done with an emphasis placed upon variables selected for the model, the importance of the variables in terms of predictive value added, and the results displayed as fit statistics calculated from a confusion matrix. The following two sections mimicked the same format but displayed model selection, variable importance and fit statistics for grade level artificial neural networks, as well as, variable selection, parameter tables, and fit statistics for grade level multilevel logistic regression models.

The final section of this chapter revisited the interpretation of the fit statistics being used for model comparison and provided an interpretation of the models' predictive success. During this process, an emphasis was placed on classification model accuracy and overall model quality (MCC). The more specific performance metrics (sensitivity, specificity, and precision) were reported as well. Model success was viewed across models / within grades, as well as, across grades / within models. In summation, it was shown that if there is a concern for overfitting of the data, artificial neural networks or multilevel logistic regression are both suitable model choices, but with the proper checks in place to stop overfitting, gradient boosted decision trees are very powerful models for successful classification. Discussions of these findings are presented in the following chapter.

CHAPTER VII: DISCUSSION

7.1 Overview

This chapter acts as a summary of the research findings, while also examining the impact they have on the field of educational analytics. Future directions for similar research are also discussed. Topics covered include the findings of the primary model comparison, secondary findings related to variables consistently predicting college enrollment across all models, solutions to data related issues like overfitting the model, and how the findings of this study can be used to guide future research.

7.2 Discussion of Primary Findings

The primary findings of the dissertation indicate that model selection should be heavily reliant on the data being analyzed. The sample that was utilized presented a lower percentage of the students enrolling in college when compared to the state average. This left the levels of classification unbalanced due to a greater number of non-event (not enrolling in college) outcomes in the training dataset. When implementing data mining models, a common downside is overfitting the model using a large number of variables, and, in turn, losing generalizability or reproducibility (Hausman, Abrevaya, & Scott-Morton, 1998). It was evident that the artificial neural networks and multilevel logistic regression did not succumb to overfitting the dataset. The gradient boosted tree model misclassified more cases than the other models on the early grades due to this issue, but once the model was trained properly, it exceeded the classification success of the other two models.

7.3 Discussion of Additional Findings

The secondary findings in this study were the discovery of variables holding high predictive value unanimously across all models and grades. The model results showed that DaysOnRoll and AttendanceRate were present on all models and highly valued. These findings, specifically the DaysOnRoll, act as a proxy for behavioral and social variables that were investigated prior to the analysis. The primary issue with the data that disallowed collapsing across all grade levels for one analysis was the inconsistency in student enrollment behavior data. The data contained an above average level of students entering and leaving the school system, entering and leaving the individual schools, receiving long term suspensions or expulsions, and dropping out during the school years. All of these behaviors can be captured when looking at DaysOnRoll. This assumption is supported by the fact that most models heavily favored DaysOnRoll as an important predictor.

During the initial exploration, it was discovered that combining all the grades would remove enough of the student level data representing the 'no higher education enrollment', it would improperly weight the data in favor of college enrollment. By removing a larger portion of the sample that almost exclusively exhibited a non-event, the certainty of training a poor model would greatly increase leading to higher rates of misclassification. The DaysOnRoll variable created a valuable snapshot of a student's overall likelihood of successfully enrolling in college simply by acting as a proxy for the underlying sources causing students to leave schools.

It was also apparent by the separation of GPA into STEM and non-STEM, that students maintaining a high STEM GPA were more likely to enter higher education than those with a low STEM GPA and high non-STEM GPA. The study also provided valuable insight into the use of the standardized tests administered by the school system like the EOI. It was also recognizable that STEM GPA was more important predictor for the data mining models than Non-STEM GPA.

The use of flag variables measuring academic intensity for STEM related courses also provided valuable binary splits for the GBDT models. These variables were not included in the HLM models due to the PROC GLMSELECT output. One primary benefit of using data mining models like GBDT or ANN is the ease at which they handle variables of any format. It became evident that the HLM models did not gain benefit when these variables were included.

In summary, the focus on variable creation focusing on specific academic behavior representing both participation in specific STEM courses and success in specific STEM courses created new and useful data that is not commonly included in statistical models. The tree structure present in a gradient boosted decision tree could successfully implement these flags and performance metrics to create more detailed splits helping predict college enrollment.

7.4 Ensemble Models

As more data is collected and utilized simultaneously, the need for models that can adequately measure outcomes and provide solutions will grow. Education is not the only domain where data creation is growing faster than data analysis. It is

important to understand that with more data, comes more potential issues with model development.

A widely used method for model development to help avoid overfitting is the use of an ensemble model. Ensemble models train many models using the same training data, but different subsets of features within the data (Oza & Tumer, 2001). These models use weighted averaging methods to combine model components and better understand the data as a whole. Mixtures of Experts methodology (Jordan & Jacobs, 1994) uses the same inputs the models were calibrated on to return an aggregate weight for each model included in the ensemble model. The weights on each model determine how much certainty the modeler has on that specific base model estimating properly (Tumer & Ghosh, 1996). The methodology is based on the assumption that if you overfit a series of models, each to a different specific subsection of the data, the models will act as a committee and properly estimate outcomes by leveraging strengths from many estimation and optimization techniques.

7.5 Future Direction

It is also important to point out that as the number of variables collected grows, it becomes increasingly difficult to rely on standard statistical methodology for applied analytic practices. The usefulness of data mining algorithms and fast, approachable ways to determine variable selection and importance will become paramount as hindrances in the field rely less on computational power and more on time. The slow adoption of data mining methodology has been due in part to the dedicated resources required to successfully store, analyze, and report on large

datasets. As technology catches up with the modeling practices and algorithms used, the value of data mining models will become more and more obvious.

Directions for future research in this area should focus on adaptive data management practice to help create streaming data inputs for data mining algorithms to calibrate to as new data is included. Automated recalibration using data as it is being collected would allow for real-time prediction and student behavior. Another area that could be investigated is the development and implementation of ensemble models to accurately predict without overfitting. Examination of other meta-algorithms (e.g. bagging and stacking) similar to the boosting algorithm used with the decision trees in this dissertation would also shed more light on what could be done to stop overfitting with educational data. Data mining models are also being trained on text data to create analyzable data out of qualitative responses. Overall, the field of data mining and machine learning is growing very fast, and it seems worthwhile to allow these models to guide the future of educational analytics.

REFERENCES

Alpaydin, E. (2011). Introduction to machine learning, 2nd ed. Cambridge, MA: MIT Press.

Ayala, G. & Yano, Y. (1998). Collaborative learning environment based on intelligent agents. Expert Systems with Applications, 14(1), 129-137.

Baeck, P. & Van den Poel, D. (2012). Including the salesperson effect in purchasing behavior using PROC GLIMMIX. Sas Global Forum 2012, (350- 2012).

Baker, S. & Yacef, K. (2009). The state of educatinal data mining in 2009: A review of future visions. Journal of Educational Data Mining. 1 (1). 4-6.

Bhise, R., Thorat, S., & Supekar, A. (2013). Importance of data mining in higher education systems. Journal of Humanities and Social Science. 6 (6). 18-20.

Birnie-Lefcovitch, S. (2000). Student perceptions of the transition from high school to university: Implications for preventative programming. Journal of the First-Year Experience and Students in Transition, 12, 61-88.

Bishop, C. (1995). Neural networks for pattern recognition. Clarendon Press, Advanced Texts in Econometrics.

Bishop, C. (2006). Pattern recognition and machine learning. Vol. 1 Springer, New York.

Blagus, R. & Lusa, L. (2017). Gradient boosting for high-dimensional prediction of rare events. Computational Statistics and Data Analysis, 113, 19-37.

Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using matthews correlation coefficient metric. PLoS ONE, 12(6).

Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123–140.

Breiman, L. (1998). Arcing classifier. The Annals of Statistics, 26(3), 801–849.

Breiman, L. (1999). Prediction games and arcing algorithms. Neural Computation, 11(7):1493–1517.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. New York, NY: Chapman & Hall.

Brewe, E., Kramer, L., & O'Brien, G. (2009). Modeling instruction: Positive attitudinal shifts in introductory physics measured with CLASS. Physical Review Physics Education Research. 5.

Brodley, C. & Friedl, M. (1999). Identifying mislabeled training data. Journal of Artificial Intelligence Research, 11, 131–167.

Byon, E., Shrivastava, A., & Ding, Y. (2010). A classification procedure for highly imbalanced class sizes. IEEE Transactions on Computers. 42. 288-303

Caelen, O. (2017). A Bayesian interpretation of the confusion matrix. Annals of Mathematics and Artificial Intelligence, 81(3), 429-450.

Cawley, G. & Talbot, N. (2007). Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. Machine Learning. 8, 841–861.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guides. SPSS. The CRISP-DM Consortium, August 2000.

Clifton, C. & Thuraisingham, B. (2001). Emerging standards for data mining. Computer Standards & Interfaces. 23 (2), 187-193.

Conley, D. (2007). Redefining college readiness. Eugene, OR: Education Policy Improvement Center.

Cristianini, N., & Shawe-Taylor, J. 2000. An Introduction to Support Vector Machines. Cambridge, U.K.: Cambrige University Press.

Crockett, K., Latham, A., & Whitton, N. (2017). On predicting learning styles in conversational intelligent tutoring systems using fuzzy decision trees. International Journal of Human-Computer Studies, 97, 98-115.

D. Pregibon (1996). Data mining, statistical computing, & graphics, Vol. 8.

Desjardins, S., & Lindsay, N. (2008). Adding a statistical wrench to the "toolbox." Research in Higher Education, 49, 172–179.

Dorian, P (1999). Data preparation for data-mining. San Francisco, Morgan Kaufmann.

Elith, J., Leathwick, J., & Hastie, T. (2008). A working guide to boosted regression trees. Journal of Animal Ecology, 77(4), 802–813.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition, 27. 861-874.

Fay, M. (2005). Random marginal agreement coefficients: Rethinking the adjustment for chance when measuring agreement. Biostatistics, 6, 171-180, 10.

Freund, Y. & Schapire, R. (1996). Experiments with a new boosting algorithm. International Conference on Machine Learning. vol. 96, 148–156.

Freund, Y. & Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5), 1189–1232

Friedman, J. (2002). Stochastic gradient boosting. Computational Statistics and Data Analysis, 38, 367–378.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. Annals of Statistics, 337–374.

Funahashi, K & Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. Neural Networks 6(6): 801-806.

Gorunescu, F. (2011). Data mining: Concepts, models and techniques. Vol. 12. Springer.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research 3:1157–1182.

Han, J., Kamber, M., & Pei, J. (2012). Data mining: Concepts and techniques. Morgan Kaufmann, San Francisco.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference and prediction, 2nd ed. Springer, New York.

Hausman, J., Abrevaya, J., & Scott-Morton, F. (1998). Misclassification of the dependent variable in a discrete-response setting. Journal of Econometrics. 87(2), 239–269.

Haykin, S. (2005). Neural networks: A comprehensive foundation (2nd ed.) Pearson Printice Hall Publication.

Haykin, S. (2008). Neural networks and learning machines. 3rd Ed. Prentice Hall Publishing, NJ.

Herrera, O. (2006). Investigation of the role of pre- and post-admission variables in undergraduate institutional persistence, using a Markov student flow model. North Carolina State University.

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feed-forward networks. Neural Networks, 3, 359-366.

Jain, A. & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence.19(2):153-158.

Jordan, M. & Jacobs, R. (1994). Hierarchical mixture of experts and the em algorithm. Neural Computation. 6, 181-214.

King, M. & Resick, P. (2014). Data mining in psychological treatment research: A primer on classification and regression trees. Journal of Consulting and Clinical Psychology. 82(5). 895-905.

King, S. (2003). Using roc curves to compare neural networks and logistic regression for modeling individual noncatastrophic tree mortality. Proceedings of the 13th Central Hardwood Forest Conference. 349-358. St. Paul, MN: U.S. Department of Agriculture.

Kohavi, R., & Provost, F. (1998). On applied research in machine learning. Applications of Machine Learning and the Knowledge Discovery Process. Vol. 30. Columbia University, New York.

Kotsiantis, S., Pierrakeas, C., & Pintelas, P. (2004). Predicting students' performance in distance learning using machine learning techniques. Applied Artificial Intelligence, 18, 411-426.

Lavrac, N. (1999). Selected techniques for data mining in medicine. Artificial Intelligence in Medicine. 16(3) 23.

Leaper, N. (2000). A visual guide to CRISP-DM methodology. CRISP-DM 1.0. http:// www.crisp-dm.org/download.htm.

Lin, W., & Chen, J. (2012). Class-imbalanced classifiers for high-dimensional data. Briefings in Bioinformatics. 14(1), 13-26.

Luan, J., & Zhao, C.. (2006). Practicing data mining for enrollment management and beyond. New Directions for Institutional Research, 31(1), 117-122.

Ludbrook, J. (2002). Statistical techniques for comparing measurers and methods of measurement: A critical review. Clinical and Experimental Pharmacology and Physiology. 29(7). 527-536.

Mackay, D. (1995). Probable networks and plausible predictions: A review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6, 469–505.

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta. 405 (2): 442–451.

Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014). The evolution of boosting algorithms: Machine learning to statistical modelling. Methods of Information in Medicine. 53, 419-427.

Mehrotra, K., Mohan, C. & Ranka, S. (1997). Elements of Artificial Neural Networks. Boston: MIT Press.

Murtaugh, P., Burns, L., & Schuster, J. (1999). Predicting the retention of university students. Research in Higher Education, 40(3), 355-371.

Neal, R. (1996). Bayesian learning for neural networks. Springer: Notes in Statistics, 118.

Nilsson, N. (1990). Learning machines: The mathematical foundations of learning machine. McGrawHill.

Nowlan, S. & Hinton, G. (1992). Simplifying neural networks by soft weight sharing. Neural Computation, 4(4), 473–493.

Olden, J. & Jackson, D. (2002). Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. Ecological Modelling, 154(1), 135–150.

Opper, M. & Winther, O. (2000). Gaussian processes for classification. Neural Computation. 12(11), 2655–2684.

Oza, N. & Tumer, K. (2001). Input decimation ensembles: Decorrelation through dimensionality reduction. Second International Workshop on Multiple Classifier Systems. Springer-Verlag. Berlin.

Perruchet, P. & Peereman, R. (2004). The exploitation of distributional information in syllable processing. Journal of Neurolinguistics. 17: 97–119.

Powers, D. (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. Journal of Machine Learning Technologies. 2 (1): 37–63.

Quinlan, J. (1986). Induction of decision trees. Machine Learning 1(1), 81–106.

Quinlan, J. (1993). C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann;

Raudenbush, S. & Bryk, A. (2012). Hierarchical linear models: Applications and data analysis methods (Advanced Quantitative Techniques in the Social Sciences). Thousand Oaks: Sage Publications.

Ripley, B. (1996). Pattern recognition and neural networks. Cambridge University Press.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagation errors. Nature, 323.

Rupp, A., Nugent, R., & Nelson, B. (2012). Evidence-centered design for diagnostic assessment within digital learning environment: Integrating modern psychometric and EDM. Journal of Educational Data Mining, 4(1), 1-10.

Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P., & Mononen, J. (2018). Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. Behavioural Processes, 148, 56-62.

Schapire, R. & Freund, Y. (2012). Boosting: Foundations and algorithms. MIT Press.

Shearer C. (2000). The CRISP-DM model: The new blueprint for data mining. Journal of Data Warehousing,(5) 13-22.

Shute, V. (1993). A comparison of learning environments: All that glitters. Lajoie, S. & Derry, S. (Eds.), Computers as Cognitive Tools (47-73). Hillsdale, NJ.

Sietsma, J. & Dow, R. (1991). Creating artificial neural networks that generalize. Neural Networks 4(1), 67–79.

Singer, J. (1998). Using SAS PROC MIXED to fit multilevel models, hierarchical models, and individual growth models. Journal of Educational and Behavioral Statistics, 24(4), 323-355.

Singh, H., Parhar, T. & Malla, S. (2015). Gesture control interface using machine learning algorithms. International Journal of Advanced Research in Computer Science and Software Engineering. 5. 898.

Siraj, F., & Abdoulha, M. (2009). Uncovering hidden information within university's student enrolment data using data mining. Journal of Computing, 1(2), 337-342.

Stoean, R., Preuss, M., Stoean, C., El-Darzi, E., & Dumitrescu, D. (2009) Support vector machine learning with an evolutionary engine. Journal of the Operational Research Society, Special Issue: Data Mining and Operational Research: Techniques and Applications 60(8), 1116–1122.

Strauss, L., & Volkwein, J. (2004). Predictors of student commitment at two- year and four-year institutions. The Journal of Higher Education, 75(2), 203-227.

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. Psychological Methods, 14(4), 323–348.

Suleiman, A., Tight, R., & Quinn, A. (2016). Hybrid neural networks and boosted regression tree models for predicting roadside particulate matter. Environmental Model Assessment. 21: 731.

Taylor, J. (1999). Neural networks and their applications. Wiley.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, 2(58), 267–288.

Ting, K. (2011). Encyclopedia of machine learning. Springer.

Tukey, J. (1977) Exploratory data analysis. Addison-Wesley: Reading.

Tumer, K. & Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. Connection Science. Special Issue on Combining Artificial Neural Networks: Ensemble Approaches, 8(3-4), 385-404.

Van de Cruys, T. (2014). A neural network approach to selectional preference acquisition. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 26–35.

Venables, W. & Ripley, B. (1997). Modern Applied Statistics with S-PLUS. New York: Springer.

Webb, A. (1994). Functional approximation by feed-forward networks: A least-squares approach to generalisation. IEEE Transactions on Neural Networks. 5(3), 363–371.

Weiss, S. & Kulikowski, C. (1991) Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. Morgan Kaufmann Publishers Inc: San Francisco.

Witten, I. & Eibe, F. (2005) Data mining: Practical machine learning tools and techniques, (2nd ed.). San Francisco

APPENDIX 1: Summary of Variables

Table 8: Variables Used in Model Development

Variable Name	Variable Description
ACTComp	Composite score on the ACT.
STEMCourseGPA	Aggregate GPA weighted by course hours from STEM specific courses. STEM course list acquired from State Regents.
NonStemCourseGPA	Aggregate GPA weighted by course hours from non-STEM specific courses.
TotalDaysOnRoll	Total days enrolled at the school or record. If more than one school of record existed, longest duration was chosen as primary school for the year.
AttendanceRate	Ratio of total days on roll and total days not absent. If more than one school of record existed, longest duration was chosen as primary school for the year.
TotalDaysSuspended	Total days suspended based on referral record.
TotalAPCoursesTaken	Combined total number of Advanced Placement courses taken in a given school year. Data was not collected for Advanced Placement credit acquired from Advanced Placement exam.
AboveGradeLevelMathCourse	Flag representing completion of Math course deemed 'Above Grade Level' for a given school year.
AboveGradeLevelScienceCourse	Flag representing completion of Science course deemed 'Above Grade Level' for a given school year.
RemedialMath	Flag representing completion of Math course deemed 'Below Grade Level' for a given school year.
RemedialScience	Flag representing completion of Science course deemed 'Below Grade Level' for a given school year.
TotalReferrals	Sum of all recorded referrals for a given school year.
EOIBiologyScore	Achievement score for the Biology portion of the EOI.
EOIAlgebraIScore	Achievement score for the Algebra I portion of the EOI.
EOIAlgebraIIScore	Achievement score for the Algebra II portion of the EOI.
EOIReadingLA2Score	Achievement score for the Reading/Language Arts II portion of the EOI.

APPENDIX 2: Artificial Neural Network Grade 9 Node Weights

Table 33: Grade 9 Artificial Neural Network Model Components

Node Path	Weights
VAR10 --> hidden neuron 1	-3.7113
VAR9 --> hidden neuron 1	0.9446
VAR8 --> hidden neuron 1	2.1668
VAR1 --> hidden neuron 1	-4.9830
VAR4 --> hidden neuron 1	-1.0209
VAR3(0) --> hidden neuron 1	5.4318
VAR3(2) --> hidden neuron 1	-2.8062
VAR3(3) --> hidden neuron 1	-1.4489
VAR3(4) --> hidden neuron 1	0.9952
VAR2(0) --> hidden neuron 1	0.0198
VAR2(1) --> hidden neuron 1	0.6043
VAR2(2) --> hidden neuron 1	4.4617
VAR2(3) --> hidden neuron 1	-0.2255
VAR2(4) --> hidden neuron 1	-2.6731
VAR5 --> hidden neuron 2	1.1770
VAR6 --> hidden neuron 2	1.6605
VAR7 --> hidden neuron 2	-1.0112
VAR10 --> hidden neuron 2	-2.2456
VAR9 --> hidden neuron 2	-1.2559
VAR8 --> hidden neuron 2	-0.0549
VAR1 --> hidden neuron 2	-5.2647
VAR4 --> hidden neuron 2	-0.6107
VAR3(0) --> hidden neuron 2	5.5039
VAR3(2) --> hidden neuron 2	-2.4222
VAR3(3) --> hidden neuron 2	-0.9593
VAR3(4) --> hidden neuron 2	0.5530
VAR2(0) --> hidden neuron 2	1.0898
VAR2(1) --> hidden neuron 2	0.4840
VAR2(2) --> hidden neuron 2	2.3911
VAR2(3) --> hidden neuron 2	-0.2891
VAR2(4) --> hidden neuron 2	-0.9311
VAR5 --> hidden neuron 3	4.6287
VAR6 --> hidden neuron 3	3.0173
VAR7 --> hidden neuron 3	0.9913
VAR10 --> hidden neuron 3	-2.2171
VAR9 --> hidden neuron 3	-11.0529
VAR8 --> hidden neuron 3	-8.4179
VAR1 --> hidden neuron 3	-1.4233
VAR4 --> hidden neuron 3	0.9305
VAR3(0) --> hidden neuron 3	3.5573
VAR3(2) --> hidden neuron 3	-4.1479

VAR3(3) --> hidden neuron 3	1.3492
VAR3(4) --> hidden neuron 3	1.4748
VAR2(0) --> hidden neuron 3	1.3150
VAR2(1) --> hidden neuron 3	1.2109
VAR2(2) --> hidden neuron 3	-2.3407
VAR2(3) --> hidden neuron 3	-0.6724
VAR2(4) --> hidden neuron 3	2.8414
VAR5 --> hidden neuron 4	-1.0090
VAR6 --> hidden neuron 4	-0.9187
VAR7 --> hidden neuron 4	-0.0018
VAR10 --> hidden neuron 4	-0.2833
VAR9 --> hidden neuron 4	3.5208
VAR8 --> hidden neuron 4	3.0476
VAR1 --> hidden neuron 4	-0.1764
VAR4 --> hidden neuron 4	-0.3191
VAR3(0) --> hidden neuron 4	0.6066
VAR3(2) --> hidden neuron 4	0.0611
VAR3(3) --> hidden neuron 4	-0.9371
VAR3(4) --> hidden neuron 4	0.3557
VAR2(0) --> hidden neuron 4	-0.7421
VAR2(1) --> hidden neuron 4	-0.1522
VAR2(2) --> hidden neuron 4	2.1423
VAR2(3) --> hidden neuron 4	0.8676
VAR2(4) --> hidden neuron 4	-1.9997
VAR5 --> hidden neuron 5	0.0431
VAR6 --> hidden neuron 5	-0.4162
VAR7 --> hidden neuron 5	1.3334
VAR10 --> hidden neuron 5	6.3835
VAR9 --> hidden neuron 5	-2.9945
VAR8 --> hidden neuron 5	-4.1668
VAR1 --> hidden neuron 5	7.9278
VAR4 --> hidden neuron 5	2.0900
VAR3(0) --> hidden neuron 5	-8.9926
VAR3(2) --> hidden neuron 5	4.6861
VAR3(3) --> hidden neuron 5	2.7849
VAR3(4) --> hidden neuron 5	-2.1977
VAR2(0) --> hidden neuron 5	-0.3521
VAR2(1) --> hidden neuron 5	-0.9894
VAR2(2) --> hidden neuron 5	-8.9200
VAR2(3) --> hidden neuron 5	0.4169
VAR2(4) --> hidden neuron 5	6.2402
VAR5 --> hidden neuron 6	-1.4563
VAR6 --> hidden neuron 6	-0.8266
VAR7 --> hidden neuron 6	0.2554
VAR10 --> hidden neuron 6	0.8228
VAR9 --> hidden neuron 6	1.7176

VAR8 --> hidden neuron 6	0.8930
VAR1 --> hidden neuron 6	1.8700
VAR4 --> hidden neuron 6	-0.0414
VAR3(0) --> hidden neuron 6	-1.8241
VAR3(2) --> hidden neuron 6	0.7321
VAR3(3) --> hidden neuron 6	0.2022
VAR3(4) --> hidden neuron 6	-0.1554
VAR2(0) --> hidden neuron 6	-0.1472
VAR2(1) --> hidden neuron 6	-0.1927
VAR2(2) --> hidden neuron 6	-0.4097
VAR2(3) --> hidden neuron 6	-0.3476
VAR2(4) --> hidden neuron 6	0.0414
VAR5 --> hidden neuron 7	4.9361
VAR6 --> hidden neuron 7	-0.4212
VAR7 --> hidden neuron 7	-2.5747
VAR10 --> hidden neuron 7	9.0361
VAR9 --> hidden neuron 7	-8.5620
VAR8 --> hidden neuron 7	-9.0929
VAR1 --> hidden neuron 7	9.7515
VAR4 --> hidden neuron 7	1.0257
VAR3(0) --> hidden neuron 7	-10.7926
VAR3(2) --> hidden neuron 7	-1.1444
VAR3(3) --> hidden neuron 7	5.0194
VAR3(4) --> hidden neuron 7	2.3863
VAR2(0) --> hidden neuron 7	4.5527
VAR2(1) --> hidden neuron 7	-0.2331
VAR2(2) --> hidden neuron 7	-2.7881
VAR2(3) --> hidden neuron 7	-16.5360
VAR2(4) --> hidden neuron 7	10.3423
VAR5 --> hidden neuron 8	0.6337
VAR6 --> hidden neuron 8	0.5663
VAR7 --> hidden neuron 8	-0.0949
VAR10 --> hidden neuron 8	0.1550
VAR9 --> hidden neuron 8	-1.9599
VAR8 --> hidden neuron 8	-1.6785
VAR1 --> hidden neuron 8	0.0807
VAR4 --> hidden neuron 8	0.2130
VAR3(0) --> hidden neuron 8	-0.2620
VAR3(2) --> hidden neuron 8	-0.2480
VAR3(3) --> hidden neuron 8	0.5289
VAR3(4) --> hidden neuron 8	-0.0781
VAR2(0) --> hidden neuron 8	0.4540
VAR2(1) --> hidden neuron 8	0.0970
VAR2(2) --> hidden neuron 8	-1.0983
VAR2(3) --> hidden neuron 8	-0.6392
VAR2(4) --> hidden neuron 8	1.1927

VAR5 --> hidden neuron 9	1.2173
VAR6 --> hidden neuron 9	0.3801
VAR7 --> hidden neuron 9	0.7556
VAR10 --> hidden neuron 9	-2.0989
VAR9 --> hidden neuron 9	0.0231
VAR8 --> hidden neuron 9	1.2346
VAR1 --> hidden neuron 9	-4.2519
VAR4 --> hidden neuron 9	-0.7552
VAR3(0) --> hidden neuron 9	4.5986
VAR3(2) --> hidden neuron 9	-0.2606
VAR3(3) --> hidden neuron 9	-1.7844
VAR3(4) --> hidden neuron 9	-0.5119
VAR2(0) --> hidden neuron 9	-0.9145
VAR2(1) --> hidden neuron 9	0.3085
VAR2(2) --> hidden neuron 9	3.0383
VAR2(3) --> hidden neuron 9	3.0809
VAR2(4) --> hidden neuron 9	-3.4706
VAR5 --> hidden neuron 10	0.5845
VAR6 --> hidden neuron 10	-0.4062
VAR7 --> hidden neuron 10	0.3643
VAR10 --> hidden neuron 10	-2.2177
VAR9 --> hidden neuron 10	1.3231
VAR8 --> hidden neuron 10	2.1932
VAR1 --> hidden neuron 10	-3.8160
VAR4 --> hidden neuron 10	-1.0039
VAR3(0) --> hidden neuron 10	4.0971
VAR3(2) --> hidden neuron 10	-0.2545
VAR3(3) --> hidden neuron 10	-1.8516
VAR3(4) --> hidden neuron 10	-0.5595
VAR2(0) --> hidden neuron 10	-1.0369
VAR2(1) --> hidden neuron 10	0.1794
VAR2(2) --> hidden neuron 10	2.6109
VAR2(3) --> hidden neuron 10	3.3441
VAR2(4) --> hidden neuron 10	-3.7763
input bias --> hidden neuron 1	2.1634
input bias --> hidden neuron 2	2.7507
input bias --> hidden neuron 3	2.2534
input bias --> hidden neuron 4	0.0710
input bias --> hidden neuron 5	-3.6475
input bias --> hidden neuron 6	-0.9706
input bias --> hidden neuron 7	-4.6110
input bias --> hidden neuron 8	-0.0466
input bias --> hidden neuron 9	2.0995
input bias --> hidden neuron 10	1.4159
hidden neuron 1 --> collegeenroll(0)	3.7549
hidden neuron 2 --> collegeenroll(0)	6.2092

hidden neuron 3 --> collegeenroll(0)	8.1521
hidden neuron 4 --> collegeenroll(0)	-1.7851
hidden neuron 5 --> collegeenroll(0)	-5.6127
hidden neuron 6 --> collegeenroll(0)	-2.0870
hidden neuron 7 --> collegeenroll(0)	-9.5121
hidden neuron 8 --> collegeenroll(0)	0.8033
hidden neuron 9 --> collegeenroll(0)	2.2795
hidden neuron 10 --> collegeenroll(0)	1.3115
hidden neuron 1 --> collegeenroll(1)	-3.7705
hidden neuron 2 --> collegeenroll(1)	-6.3005
hidden neuron 3 --> collegeenroll(1)	-8.1280
hidden neuron 4 --> collegeenroll(1)	1.7525
hidden neuron 5 --> collegeenroll(1)	5.6378
hidden neuron 6 --> collegeenroll(1)	2.0779
hidden neuron 7 --> collegeenroll(1)	9.4742
hidden neuron 8 --> collegeenroll(1)	-0.8334
hidden neuron 9 --> collegeenroll(1)	-2.2062
hidden neuron 10 --> collegeenroll(1)	-1.3082
hidden bias --> collegeenroll(0)	3.3896
hidden bias --> collegeenroll(1)	-3.4067

APPENDIX 3: Artificial Neural Network Grade 10 Node Weights

Table 34: Grade 10 Artificial Neural Network Model Components

Node Path	Weights
VAR5 --> hidden neuron 1	2.66158
VAR6 --> hidden neuron 1	-0.79704
VAR7 --> hidden neuron 1	-0.70727
VAR10 --> hidden neuron 1	-0.71177
VAR9 --> hidden neuron 1	2.48436
VAR8 --> hidden neuron 1	-0.99247
VAR12 --> hidden neuron 1	0.80641
VAR13 --> hidden neuron 1	-1.45253
VAR14 --> hidden neuron 1	0.29536
VAR1 --> hidden neuron 1	2.59977
VAR11 --> hidden neuron 1	-1.98856
VAR4 --> hidden neuron 1	-0.05522
VAR2(0) --> hidden neuron 1	0.66557
VAR2(1) --> hidden neuron 1	-0.78808
VAR2(2) --> hidden neuron 1	-0.53411
VAR2(3) --> hidden neuron 1	0.17379
VAR2(4) --> hidden neuron 1	-0.02219
VAR3(0) --> hidden neuron 1	-0.26886
VAR3(1) --> hidden neuron 1	-0.51685
VAR3(2) --> hidden neuron 1	-0.18363
VAR3(3) --> hidden neuron 1	-0.01559
VAR3(4) --> hidden neuron 1	0.46292
VAR5 --> hidden neuron 2	-0.12163
VAR6 --> hidden neuron 2	-0.42163
VAR7 --> hidden neuron 2	-0.50920
VAR10 --> hidden neuron 2	1.26436
VAR9 --> hidden neuron 2	-0.72258
VAR8 --> hidden neuron 2	-2.22553
VAR12 --> hidden neuron 2	-0.79047
VAR13 --> hidden neuron 2	-1.51759
VAR14 --> hidden neuron 2	0.88944
VAR1 --> hidden neuron 2	0.46894
VAR11 --> hidden neuron 2	0.43835
VAR4 --> hidden neuron 2	-0.35146
VAR2(0) --> hidden neuron 2	0.52920
VAR2(1) --> hidden neuron 2	0.52606
VAR2(2) --> hidden neuron 2	-0.71318
VAR2(3) --> hidden neuron 2	0.24217
VAR2(4) --> hidden neuron 2	-0.10640
VAR3(0) --> hidden neuron 2	0.38833
VAR3(1) --> hidden neuron 2	0.46626

VAR3(2) --> hidden neuron 2	0.59880
VAR3(3) --> hidden neuron 2	-0.31790
VAR3(4) --> hidden neuron 2	-0.62235
VAR5 --> hidden neuron 3	-0.21500
VAR6 --> hidden neuron 3	0.05185
VAR7 --> hidden neuron 3	-0.28822
VAR10 --> hidden neuron 3	-0.12314
VAR9 --> hidden neuron 3	0.01955
VAR8 --> hidden neuron 3	0.92717
VAR12 --> hidden neuron 3	-1.90841
VAR13 --> hidden neuron 3	-2.25807
VAR14 --> hidden neuron 3	0.91948
VAR1 --> hidden neuron 3	2.56592
VAR11 --> hidden neuron 3	0.40869
VAR4 --> hidden neuron 3	0.55210
VAR2(0) --> hidden neuron 3	-1.19653
VAR2(1) --> hidden neuron 3	-0.62203
VAR2(2) --> hidden neuron 3	0.50102
VAR2(3) --> hidden neuron 3	0.56936
VAR2(4) --> hidden neuron 3	0.58012
VAR3(0) --> hidden neuron 3	1.31829
VAR3(1) --> hidden neuron 3	-2.15567
VAR3(2) --> hidden neuron 3	0.44430
VAR3(3) --> hidden neuron 3	0.54150
VAR3(4) --> hidden neuron 3	-0.24054
VAR5 --> hidden neuron 4	-0.49716
VAR6 --> hidden neuron 4	-3.32836
VAR7 --> hidden neuron 4	0.96439
VAR10 --> hidden neuron 4	3.53210
VAR9 --> hidden neuron 4	-2.61825
VAR8 --> hidden neuron 4	-1.68655
VAR12 --> hidden neuron 4	1.18990
VAR13 --> hidden neuron 4	0.82669
VAR14 --> hidden neuron 4	-0.75523
VAR1 --> hidden neuron 4	-0.52751
VAR11 --> hidden neuron 4	0.48492
VAR4 --> hidden neuron 4	0.72920
VAR2(0) --> hidden neuron 4	-1.32038
VAR2(1) --> hidden neuron 4	-0.79756
VAR2(2) --> hidden neuron 4	-1.54595
VAR2(3) --> hidden neuron 4	0.43585
VAR2(4) --> hidden neuron 4	1.93924
VAR3(0) --> hidden neuron 4	-0.06176
VAR3(1) --> hidden neuron 4	-0.91869
VAR3(2) --> hidden neuron 4	-0.02668
VAR3(3) --> hidden neuron 4	-0.83516

VAR3(4) --> hidden neuron 4	0.57113
VAR5 --> hidden neuron 5	1.54219
VAR6 --> hidden neuron 5	-2.04265
VAR7 --> hidden neuron 5	0.26241
VAR10 --> hidden neuron 5	1.22997
VAR9 --> hidden neuron 5	0.89092
VAR8 --> hidden neuron 5	-1.18849
VAR12 --> hidden neuron 5	1.31068
VAR13 --> hidden neuron 5	-0.41383
VAR14 --> hidden neuron 5	1.42727
VAR1 --> hidden neuron 5	-1.42726
VAR11 --> hidden neuron 5	-2.06292
VAR4 --> hidden neuron 5	0.12135
VAR2(0) --> hidden neuron 5	-0.80572
VAR2(1) --> hidden neuron 5	-0.85567
VAR2(2) --> hidden neuron 5	-0.92655
VAR2(3) --> hidden neuron 5	0.39048
VAR2(4) --> hidden neuron 5	0.89146
VAR3(0) --> hidden neuron 5	1.38915
VAR3(1) --> hidden neuron 5	-2.56693
VAR3(2) --> hidden neuron 5	-0.00357
VAR3(3) --> hidden neuron 5	-0.24807
VAR3(4) --> hidden neuron 5	0.22878
VAR5 --> hidden neuron 6	-0.46279
VAR6 --> hidden neuron 6	-0.76762
VAR7 --> hidden neuron 6	-1.09629
VAR10 --> hidden neuron 6	1.85300
VAR9 --> hidden neuron 6	-0.09270
VAR8 --> hidden neuron 6	-3.10167
VAR12 --> hidden neuron 6	-0.12832
VAR13 --> hidden neuron 6	-2.23603
VAR14 --> hidden neuron 6	-0.48361
VAR1 --> hidden neuron 6	0.26868
VAR11 --> hidden neuron 6	0.91823
VAR4 --> hidden neuron 6	-0.59016
VAR2(0) --> hidden neuron 6	0.38600
VAR2(1) --> hidden neuron 6	0.14059
VAR2(2) --> hidden neuron 6	0.51270
VAR2(3) --> hidden neuron 6	0.49486
VAR2(4) --> hidden neuron 6	-1.49155
VAR3(0) --> hidden neuron 6	-0.10660
VAR3(1) --> hidden neuron 6	0.53724
VAR3(2) --> hidden neuron 6	-0.22643
VAR3(3) --> hidden neuron 6	0.16762
VAR3(4) --> hidden neuron 6	-0.39447
input bias --> hidden neuron 1	-0.49102

input bias --> hidden neuron 2	0.52416
input bias --> hidden neuron 3	-0.14994
input bias --> hidden neuron 4	-1.31502
input bias --> hidden neuron 5	-1.24418
input bias --> hidden neuron 6	-0.00149
hidden neuron 1 --> collegeenroll(0)	0.29624
hidden neuron 2 --> collegeenroll(0)	0.10854
hidden neuron 3 --> collegeenroll(0)	0.44195
hidden neuron 4 --> collegeenroll(0)	0.28873
hidden neuron 5 --> collegeenroll(0)	-0.83638
hidden neuron 6 --> collegeenroll(0)	0.91895
hidden neuron 1 --> collegeenroll(1)	-0.27805
hidden neuron 2 --> collegeenroll(1)	-0.12433
hidden neuron 3 --> collegeenroll(1)	-0.45000
hidden neuron 4 --> collegeenroll(1)	-0.28072
hidden neuron 5 --> collegeenroll(1)	0.80305
hidden neuron 6 --> collegeenroll(1)	-0.75844
hidden bias --> collegeenroll(0)	-0.50774
hidden bias --> collegeenroll(1)	1.49524
VAR5 --> hidden neuron 1	2.66158
VAR6 --> hidden neuron 1	-0.79704
VAR7 --> hidden neuron 1	-0.70727
VAR10 --> hidden neuron 1	-0.71177
VAR9 --> hidden neuron 1	2.48436
VAR8 --> hidden neuron 1	-0.99247
VAR12 --> hidden neuron 1	0.80641
VAR13 --> hidden neuron 1	-1.45253
VAR14 --> hidden neuron 1	0.29536
VAR1 --> hidden neuron 1	2.59977
VAR11 --> hidden neuron 1	-1.98856
VAR4 --> hidden neuron 1	-0.05522
VAR2(0) --> hidden neuron 1	0.66557
VAR2(1) --> hidden neuron 1	-0.78808
VAR2(2) --> hidden neuron 1	-0.53411
VAR2(3) --> hidden neuron 1	0.17379
VAR2(4) --> hidden neuron 1	-0.02219
VAR3(0) --> hidden neuron 1	-0.26886
VAR3(1) --> hidden neuron 1	-0.51685
VAR3(2) --> hidden neuron 1	-0.18363
VAR3(3) --> hidden neuron 1	-0.01559
VAR3(4) --> hidden neuron 1	0.46292
VAR5 --> hidden neuron 2	-0.12163
VAR6 --> hidden neuron 2	-0.42163
VAR7 --> hidden neuron 2	-0.50920
VAR10 --> hidden neuron 2	1.26436
VAR9 --> hidden neuron 2	-0.72258

VAR8 --> hidden neuron 2	-2.22553
VAR12 --> hidden neuron 2	-0.79047
VAR13 --> hidden neuron 2	-1.51759
VAR14 --> hidden neuron 2	0.88944
VAR1 --> hidden neuron 2	0.46894
VAR11 --> hidden neuron 2	0.43835
VAR4 --> hidden neuron 2	-0.35146
VAR2(0) --> hidden neuron 2	0.52920
VAR2(1) --> hidden neuron 2	0.52606
VAR2(2) --> hidden neuron 2	-0.71318
VAR2(3) --> hidden neuron 2	0.24217
VAR2(4) --> hidden neuron 2	-0.10640

APPENDIX 4: Artificial Neural Network Grade 11 Node Weights

Table 35: Grade 11 Artificial Neural Network Model Components

Node Path	Weights
VAR5 --> hidden neuron 1	-1.07981
VAR6 --> hidden neuron 1	-0.16278
VAR7 --> hidden neuron 1	-0.31475
VAR10 --> hidden neuron 1	0.22533
VAR9 --> hidden neuron 1	0.52433
VAR8 --> hidden neuron 1	-0.06852
VAR12 --> hidden neuron 1	-0.75497
VAR13 --> hidden neuron 1	0.01439
VAR1 --> hidden neuron 1	-0.28976
VAR11 --> hidden neuron 1	1.01031
VAR4 --> hidden neuron 1	-0.11432
VAR3(0) --> hidden neuron 1	1.30022
VAR3(1) --> hidden neuron 1	-0.26916
VAR3(2) --> hidden neuron 1	-0.12863
VAR3(3) --> hidden neuron 1	0.38414
VAR3(4) --> hidden neuron 1	-0.37746
VAR2(0) --> hidden neuron 1	1.04106
VAR2(1) --> hidden neuron 1	-0.60659
VAR2(2) --> hidden neuron 1	1.18126
VAR2(3) --> hidden neuron 1	-1.07853
VAR2(4) --> hidden neuron 1	0.37134
VAR5 --> hidden neuron 2	-0.31864
VAR6 --> hidden neuron 2	0.09271
VAR7 --> hidden neuron 2	-0.16563
VAR10 --> hidden neuron 2	0.12617
VAR9 --> hidden neuron 2	0.14468
VAR8 --> hidden neuron 2	-0.07808
VAR12 --> hidden neuron 2	-0.32457
VAR13 --> hidden neuron 2	0.03525
VAR1 --> hidden neuron 2	-0.02018
VAR11 --> hidden neuron 2	0.42364
VAR4 --> hidden neuron 2	-0.00690
VAR3(0) --> hidden neuron 2	0.39664
VAR3(1) --> hidden neuron 2	-0.04808
VAR3(2) --> hidden neuron 2	0.01147
VAR3(3) --> hidden neuron 2	-0.16130
VAR3(4) --> hidden neuron 2	0.05324
VAR2(0) --> hidden neuron 2	0.11118
VAR2(1) --> hidden neuron 2	0.11949
VAR2(2) --> hidden neuron 2	-0.20212
VAR2(3) --> hidden neuron 2	0.11232

VAR2(4) --> hidden neuron 2	0.07257
VAR5 --> hidden neuron 3	-0.11655
VAR6 --> hidden neuron 3	-0.03049
VAR7 --> hidden neuron 3	-0.02406
VAR10 --> hidden neuron 3	0.00606
VAR9 --> hidden neuron 3	0.00375
VAR8 --> hidden neuron 3	-0.03738
VAR12 --> hidden neuron 3	-0.06968
VAR13 --> hidden neuron 3	-0.04132
VAR1 --> hidden neuron 3	-0.06708
VAR11 --> hidden neuron 3	0.08576
VAR4 --> hidden neuron 3	-0.05293
VAR3(0) --> hidden neuron 3	0.06111
VAR3(1) --> hidden neuron 3	-0.03256
VAR3(2) --> hidden neuron 3	-0.02006
VAR3(3) --> hidden neuron 3	0.04535
VAR3(4) --> hidden neuron 3	-0.01279
VAR2(0) --> hidden neuron 3	0.02330
VAR2(1) --> hidden neuron 3	-0.06067
VAR2(2) --> hidden neuron 3	0.04449
VAR2(3) --> hidden neuron 3	-0.04886
VAR2(4) --> hidden neuron 3	0.00079
VAR5 --> hidden neuron 4	-1.19733
VAR6 --> hidden neuron 4	-0.53499
VAR7 --> hidden neuron 4	-0.43106
VAR10 --> hidden neuron 4	-0.15906
VAR9 --> hidden neuron 4	0.48030
VAR8 --> hidden neuron 4	-0.08219
VAR12 --> hidden neuron 4	-0.83030
VAR13 --> hidden neuron 4	-0.12991
VAR1 --> hidden neuron 4	-0.49328
VAR11 --> hidden neuron 4	0.77075
VAR4 --> hidden neuron 4	-0.46354
VAR3(0) --> hidden neuron 4	1.01958
VAR3(1) --> hidden neuron 4	-0.03023
VAR3(2) --> hidden neuron 4	-0.02757
VAR3(3) --> hidden neuron 4	0.36595
VAR3(4) --> hidden neuron 4	-0.91769
VAR2(0) --> hidden neuron 4	0.94344
VAR2(1) --> hidden neuron 4	-0.60914
VAR2(2) --> hidden neuron 4	1.28338
VAR2(3) --> hidden neuron 4	-1.39848
VAR2(4) --> hidden neuron 4	0.17139
VAR5 --> hidden neuron 5	-0.66299
VAR6 --> hidden neuron 5	-0.15425
VAR7 --> hidden neuron 5	-0.42444

VAR10 --> hidden neuron 5	-0.02554
VAR9 --> hidden neuron 5	0.16188
VAR8 --> hidden neuron 5	-0.06941
VAR12 --> hidden neuron 5	-0.65968
VAR13 --> hidden neuron 5	-0.14041
VAR1 --> hidden neuron 5	-0.31107
VAR11 --> hidden neuron 5	0.50708
VAR4 --> hidden neuron 5	-0.14116
VAR3(0) --> hidden neuron 5	0.65308
VAR3(1) --> hidden neuron 5	-0.21464
VAR3(2) --> hidden neuron 5	-0.12706
VAR3(3) --> hidden neuron 5	-0.02010
VAR3(4) --> hidden neuron 5	-0.05170
VAR2(0) --> hidden neuron 5	0.37898
VAR2(1) --> hidden neuron 5	-0.22253
VAR2(2) --> hidden neuron 5	0.24792
VAR2(3) --> hidden neuron 5	-0.16448
VAR2(4) --> hidden neuron 5	-0.00649
VAR5 --> hidden neuron 6	-0.34152
VAR6 --> hidden neuron 6	-0.22134
VAR7 --> hidden neuron 6	0.01140
VAR10 --> hidden neuron 6	0.03320
VAR9 --> hidden neuron 6	0.21415
VAR8 --> hidden neuron 6	0.01101
VAR12 --> hidden neuron 6	-0.17869
VAR13 --> hidden neuron 6	-0.02157
VAR1 --> hidden neuron 6	-0.16434
VAR11 --> hidden neuron 6	0.20839
VAR4 --> hidden neuron 6	-0.12370
VAR3(0) --> hidden neuron 6	0.27842
VAR3(1) --> hidden neuron 6	0.08062
VAR3(2) --> hidden neuron 6	-0.01435
VAR3(3) --> hidden neuron 6	0.30525
VAR3(4) --> hidden neuron 6	-0.43290
VAR2(0) --> hidden neuron 6	0.41275
VAR2(1) --> hidden neuron 6	-0.33925
VAR2(2) --> hidden neuron 6	0.68638
VAR2(3) --> hidden neuron 6	-0.77970
VAR2(4) --> hidden neuron 6	0.15972
VAR5 --> hidden neuron 7	-0.79641
VAR6 --> hidden neuron 7	-0.05358
VAR7 --> hidden neuron 7	-0.44325
VAR10 --> hidden neuron 7	0.09005
VAR9 --> hidden neuron 7	0.26565
VAR8 --> hidden neuron 7	-0.06534
VAR12 --> hidden neuron 7	-0.78485

VAR13 --> hidden neuron 7	-0.10882
VAR1 --> hidden neuron 7	-0.29348
VAR11 --> hidden neuron 7	0.73727
VAR4 --> hidden neuron 7	-0.05348
VAR3(0) --> hidden neuron 7	0.79497
VAR3(1) --> hidden neuron 7	-0.39418
VAR3(2) --> hidden neuron 7	-0.17158
VAR3(3) --> hidden neuron 7	0.01573
VAR3(4) --> hidden neuron 7	-0.00792
VAR2(0) --> hidden neuron 7	0.38290
VAR2(1) --> hidden neuron 7	-0.33666
VAR2(2) --> hidden neuron 7	0.20344
VAR2(3) --> hidden neuron 7	-0.00780
VAR2(4) --> hidden neuron 7	0.07402
VAR5 --> hidden neuron 8	-0.43114
VAR6 --> hidden neuron 8	0.01018
VAR7 --> hidden neuron 8	-0.13467
VAR10 --> hidden neuron 8	0.08983
VAR9 --> hidden neuron 8	0.24879
VAR8 --> hidden neuron 8	-0.00927
VAR12 --> hidden neuron 8	-0.41035
VAR13 --> hidden neuron 8	0.00854
VAR1 --> hidden neuron 8	-0.17446
VAR11 --> hidden neuron 8	0.42310
VAR4 --> hidden neuron 8	-0.07470
VAR3(0) --> hidden neuron 8	0.46528
VAR3(1) --> hidden neuron 8	-0.06815
VAR3(2) --> hidden neuron 8	0.01312
VAR3(3) --> hidden neuron 8	0.06115
VAR3(4) --> hidden neuron 8	-0.15042
VAR2(0) --> hidden neuron 8	0.32958
VAR2(1) --> hidden neuron 8	-0.03417
VAR2(2) --> hidden neuron 8	0.27126
VAR2(3) --> hidden neuron 8	-0.22344
VAR2(4) --> hidden neuron 8	0.03806
VAR5 --> hidden neuron 9	0.40075
VAR6 --> hidden neuron 9	0.22609
VAR7 --> hidden neuron 9	0.08698
VAR10 --> hidden neuron 9	0.16581
VAR9 --> hidden neuron 9	-0.22677
VAR8 --> hidden neuron 9	0.03096
VAR12 --> hidden neuron 9	0.23325
VAR13 --> hidden neuron 9	0.12274
VAR1 --> hidden neuron 9	0.21288
VAR11 --> hidden neuron 9	-0.13223
VAR4 --> hidden neuron 9	0.28706

VAR3(0) --> hidden neuron 9	-0.16327
VAR3(2) --> hidden neuron 9	-0.01442
VAR3(3) --> hidden neuron 9	-0.01156
VAR3(4) --> hidden neuron 9	0.46847
VAR2(0) --> hidden neuron 9	-0.43979
VAR2(1) --> hidden neuron 9	0.32863
VAR2(2) --> hidden neuron 9	-0.36551
VAR2(3) --> hidden neuron 9	0.79021
VAR2(4) --> hidden neuron 9	-0.37658
VAR5 --> hidden neuron 10	0.83743
VAR6 --> hidden neuron 10	0.26761
VAR7 --> hidden neuron 10	0.29605
VAR10 --> hidden neuron 10	-0.00569
VAR9 --> hidden neuron 10	-0.31839
VAR8 --> hidden neuron 10	0.05739
VAR12 --> hidden neuron 10	0.62713
VAR13 --> hidden neuron 10	0.13553
VAR1 --> hidden neuron 10	0.35418
VAR11 --> hidden neuron 10	-0.55342
VAR4 --> hidden neuron 10	0.22281
VAR3(0) --> hidden neuron 10	-0.77364
VAR3(1) --> hidden neuron 10	0.12978
VAR3(2) --> hidden neuron 10	0.04941
VAR3(3) --> hidden neuron 10	-0.28437
VAR3(4) --> hidden neuron 10	0.40029
VAR2(0) --> hidden neuron 10	-0.66810
VAR2(1) --> hidden neuron 10	0.40443
VAR2(2) --> hidden neuron 10	-0.84134
VAR2(3) --> hidden neuron 10	0.75282
VAR2(4) --> hidden neuron 10	-0.11464
VAR5 --> hidden neuron 11	-0.01354
VAR6 --> hidden neuron 11	0.06217
VAR7 --> hidden neuron 11	-0.02758
VAR10 --> hidden neuron 11	0.00650
VAR9 --> hidden neuron 11	0.00482
VAR8 --> hidden neuron 11	-0.03592
VAR12 --> hidden neuron 11	-0.03104
VAR13 --> hidden neuron 11	0.03815
VAR1 --> hidden neuron 11	0.01217
VAR11 --> hidden neuron 11	0.06566
VAR4 --> hidden neuron 11	0.01473
VAR3(0) --> hidden neuron 11	0.00756
VAR3(1) --> hidden neuron 11	-0.05023
VAR3(2) --> hidden neuron 11	0.00353
VAR3(3) --> hidden neuron 11	-0.09358
VAR3(4) --> hidden neuron 11	0.07096

VAR2(0) --> hidden neuron 11	-0.00959
VAR2(1) --> hidden neuron 11	0.05716
VAR2(2) --> hidden neuron 11	-0.15015
VAR2(3) --> hidden neuron 11	0.14540
VAR2(4) --> hidden neuron 11	0.01020
VAR5 --> hidden neuron 12	-0.12051
VAR6 --> hidden neuron 12	0.05800
VAR7 --> hidden neuron 12	-0.09004
VAR10 --> hidden neuron 12	0.06353
VAR9 --> hidden neuron 12	0.04132
VAR8 --> hidden neuron 12	-0.02035
VAR12 --> hidden neuron 12	-0.11904
VAR13 --> hidden neuron 12	0.02682
VAR1 --> hidden neuron 12	-0.01966
VAR11 --> hidden neuron 12	0.08887
VAR4 --> hidden neuron 12	0.02777
VAR3(0) --> hidden neuron 12	0.10262
VAR3(1) --> hidden neuron 12	-0.04978
VAR3(2) --> hidden neuron 12	0.03131
VAR3(3) --> hidden neuron 12	-0.14649
VAR3(4) --> hidden neuron 12	0.03504
VAR2(0) --> hidden neuron 12	-0.05454
VAR2(1) --> hidden neuron 12	0.08721
VAR2(2) --> hidden neuron 12	-0.17393
VAR2(3) --> hidden neuron 12	0.10863
VAR2(4) --> hidden neuron 12	0.00760
input bias --> hidden neuron 1	0.93543
input bias --> hidden neuron 2	0.24165
input bias --> hidden neuron 3	-0.02542
input bias --> hidden neuron 4	0.48007
input bias --> hidden neuron 5	0.21224
input bias --> hidden neuron 6	0.22919
input bias --> hidden neuron 7	0.30596
input bias --> hidden neuron 8	0.35936
input bias --> hidden neuron 9	0.04040
input bias --> hidden neuron 10	-0.43470
input bias --> hidden neuron 11	0.05897
input bias --> hidden neuron 12	0.02339
hidden neuron 1 --> collegeenroll(0)	0.89139
hidden neuron 2 --> collegeenroll(0)	1.50458
hidden neuron 3 --> collegeenroll(0)	0.11996
hidden neuron 4 --> collegeenroll(0)	-0.34762
hidden neuron 5 --> collegeenroll(0)	1.19259
hidden neuron 6 --> collegeenroll(0)	-0.81720
hidden neuron 7 --> collegeenroll(0)	1.74305
hidden neuron 8 --> collegeenroll(0)	0.83660

hidden neuron 9 --> collegeenroll(0)	1.37894
hidden neuron 10 --> collegeenroll(0)	-0.43688
hidden neuron 11 --> collegeenroll(0)	0.43419
hidden neuron 12 --> collegeenroll(0)	0.66300
hidden neuron 1 --> collegeenroll(1)	-0.86380
hidden neuron 2 --> collegeenroll(1)	-1.44151
hidden neuron 3 --> collegeenroll(1)	-0.10134
hidden neuron 4 --> collegeenroll(1)	0.38616
hidden neuron 5 --> collegeenroll(1)	-1.21541
hidden neuron 6 --> collegeenroll(1)	0.84581
hidden neuron 7 --> collegeenroll(1)	-1.71709
hidden neuron 8 --> collegeenroll(1)	-0.80667
hidden neuron 9 --> collegeenroll(1)	-1.42701
hidden neuron 10 --> collegeenroll(1)	0.43715
hidden neuron 11 --> collegeenroll(1)	-0.43587
hidden neuron 12 --> collegeenroll(1)	-0.67763
hidden bias --> collegeenroll(0)	0.67833
hidden bias --> collegeenroll(1)	-0.72367

APPENDIX 5: Artificial Neural Network Grade 12 Node Weights

Table 36: Grade 12 Artificial Neural Network Model Components

Node Path	Weights
VAR5 --> hidden neuron 1	-2.4271
VAR6 --> hidden neuron 1	-10.2395
VAR7 --> hidden neuron 1	3.1569
VAR10 --> hidden neuron 1	-1.2375
VAR9 --> hidden neuron 1	-0.4375
VAR8 --> hidden neuron 1	-0.5695
VAR12 --> hidden neuron 1	-6.4106
VAR13 --> hidden neuron 1	11.8356
VAR1 --> hidden neuron 1	-1.6717
VAR11 --> hidden neuron 1	-3.2228
VAR4 --> hidden neuron 1	-1.7481
VAR2(0) --> hidden neuron 1	0.7098
VAR2(1) --> hidden neuron 1	0.8935
VAR2(2) --> hidden neuron 1	7.0309
VAR2(3) --> hidden neuron 1	-6.8142
VAR2(4) --> hidden neuron 1	-3.3450
VAR3(0) --> hidden neuron 1	-5.0688
VAR3(1) --> hidden neuron 1	4.2943
VAR3(2) --> hidden neuron 1	-3.3745
VAR3(3) --> hidden neuron 1	2.6952
VAR3(4) --> hidden neuron 1	-0.0490
VAR5 --> hidden neuron 2	-2.7425
VAR6 --> hidden neuron 2	-0.7160
VAR7 --> hidden neuron 2	6.6629
VAR10 --> hidden neuron 2	4.1105
VAR9 --> hidden neuron 2	1.1869
VAR8 --> hidden neuron 2	-1.3314
VAR12 --> hidden neuron 2	-0.5877
VAR13 --> hidden neuron 2	0.7232
VAR1 --> hidden neuron 2	-4.1035
VAR11 --> hidden neuron 2	1.7972
VAR4 --> hidden neuron 2	-0.3031
VAR2(0) --> hidden neuron 2	3.1302
VAR2(1) --> hidden neuron 2	-3.9807
VAR2(2) --> hidden neuron 2	3.3228
VAR2(3) --> hidden neuron 2	-2.2593
VAR2(4) --> hidden neuron 2	1.6530
VAR3(0) --> hidden neuron 2	3.2038
VAR3(1) --> hidden neuron 2	3.2815
VAR3(2) --> hidden neuron 2	-5.5505
VAR3(3) --> hidden neuron 2	0.5486

VAR3(4) --> hidden neuron 2	0.2870
VAR5 --> hidden neuron 3	-3.1645
VAR6 --> hidden neuron 3	2.0882
VAR7 --> hidden neuron 3	-2.0796
VAR10 --> hidden neuron 3	-1.1709
VAR9 --> hidden neuron 3	0.7906
VAR8 --> hidden neuron 3	1.1038
VAR12 --> hidden neuron 3	-0.0380
VAR13 --> hidden neuron 3	1.1824
VAR1 --> hidden neuron 3	1.5225
VAR11 --> hidden neuron 3	-3.9238
VAR4 --> hidden neuron 3	1.2052
VAR2(0) --> hidden neuron 3	2.6315
VAR2(1) --> hidden neuron 3	-0.1310
VAR2(2) --> hidden neuron 3	0.3420
VAR2(3) --> hidden neuron 3	0.2160
VAR2(4) --> hidden neuron 3	1.3613
VAR3(0) --> hidden neuron 3	1.6898
VAR3(1) --> hidden neuron 3	3.8221
VAR3(2) --> hidden neuron 3	-3.2575
VAR3(3) --> hidden neuron 3	3.2081
VAR3(4) --> hidden neuron 3	-0.9738
VAR5 --> hidden neuron 4	4.3464
VAR6 --> hidden neuron 4	7.6987
VAR7 --> hidden neuron 4	-4.9896
VAR10 --> hidden neuron 4	-4.2760
VAR9 --> hidden neuron 4	6.9124
VAR8 --> hidden neuron 4	-2.8507
VAR12 --> hidden neuron 4	7.4019
VAR13 --> hidden neuron 4	9.3004
VAR1 --> hidden neuron 4	0.3200
VAR11 --> hidden neuron 4	-5.3396
VAR4 --> hidden neuron 4	-1.6297
VAR2(0) --> hidden neuron 4	0.0970
VAR2(1) --> hidden neuron 4	-6.1430
VAR2(2) --> hidden neuron 4	5.6063
VAR2(3) --> hidden neuron 4	-3.7962
VAR2(4) --> hidden neuron 4	2.9452
VAR3(0) --> hidden neuron 4	4.1177
VAR3(1) --> hidden neuron 4	1.2148
VAR3(2) --> hidden neuron 4	-0.7250
VAR3(3) --> hidden neuron 4	-5.5064
VAR3(4) --> hidden neuron 4	-0.4630
VAR5 --> hidden neuron 5	4.3810
VAR6 --> hidden neuron 5	-6.4053
VAR7 --> hidden neuron 5	32.4178

VAR10 --> hidden neuron 5	-2.3202
VAR9 --> hidden neuron 5	2.9221
VAR8 --> hidden neuron 5	0.4341
VAR12 --> hidden neuron 5	-7.4284
VAR13 --> hidden neuron 5	2.3537
VAR1 --> hidden neuron 5	4.4765
VAR11 --> hidden neuron 5	-3.6527
VAR4 --> hidden neuron 5	-4.8914
VAR2(0) --> hidden neuron 5	-9.8154
VAR2(1) --> hidden neuron 5	-3.6333
VAR2(2) --> hidden neuron 5	-3.2475
VAR2(3) --> hidden neuron 5	-0.3456
VAR2(4) --> hidden neuron 5	0.9681
VAR3(0) --> hidden neuron 5	-2.7657
VAR3(1) --> hidden neuron 5	-4.0043
VAR3(2) --> hidden neuron 5	-3.1207
VAR3(3) --> hidden neuron 5	-2.9561
VAR3(4) --> hidden neuron 5	-3.2059
VAR5 --> hidden neuron 6	3.5439
VAR6 --> hidden neuron 6	-3.6116
VAR7 --> hidden neuron 6	1.2286
VAR10 --> hidden neuron 6	4.9381
VAR9 --> hidden neuron 6	10.2706
VAR8 --> hidden neuron 6	-1.0204
VAR12 --> hidden neuron 6	-0.3155
VAR13 --> hidden neuron 6	2.0615
VAR1 --> hidden neuron 6	-1.9071
VAR11 --> hidden neuron 6	-6.4048
VAR4 --> hidden neuron 6	-0.6361
VAR2(0) --> hidden neuron 6	-2.2844
VAR2(1) --> hidden neuron 6	-1.4419
VAR2(2) --> hidden neuron 6	-1.1444
VAR2(3) --> hidden neuron 6	-0.0568
VAR2(4) --> hidden neuron 6	0.2860
VAR3(0) --> hidden neuron 6	-2.9769
VAR3(1) --> hidden neuron 6	-6.1172
VAR3(2) --> hidden neuron 6	3.8552
VAR3(3) --> hidden neuron 6	5.8795
VAR3(4) --> hidden neuron 6	-5.3997
VAR5 --> hidden neuron 7	3.1596
VAR6 --> hidden neuron 7	-5.4269
VAR7 --> hidden neuron 7	-2.5966
VAR10 --> hidden neuron 7	1.8031
VAR9 --> hidden neuron 7	4.3552
VAR8 --> hidden neuron 7	-0.6169
VAR12 --> hidden neuron 7	6.0163

VAR13 --> hidden neuron 7	-1.2849
VAR1 --> hidden neuron 7	6.2791
VAR11 --> hidden neuron 7	-1.9047
VAR4 --> hidden neuron 7	0.5574
VAR2(0) --> hidden neuron 7	-2.6596
VAR2(1) --> hidden neuron 7	-1.1806
VAR2(2) --> hidden neuron 7	-1.2486
VAR2(3) --> hidden neuron 7	1.3847
VAR2(4) --> hidden neuron 7	-0.1704
VAR3(0) --> hidden neuron 7	-5.6939
VAR3(1) --> hidden neuron 7	5.3688
VAR3(2) --> hidden neuron 7	-3.0806
VAR3(3) --> hidden neuron 7	-7.1774
VAR3(4) --> hidden neuron 7	6.7051
VAR5 --> hidden neuron 8	-3.2436
VAR6 --> hidden neuron 8	1.3813
VAR7 --> hidden neuron 8	-5.5919
VAR10 --> hidden neuron 8	-3.1332
VAR9 --> hidden neuron 8	1.2757
VAR8 --> hidden neuron 8	1.7340
VAR12 --> hidden neuron 8	-0.8385
VAR13 --> hidden neuron 8	-2.1050
VAR1 --> hidden neuron 8	0.3353
VAR11 --> hidden neuron 8	1.2490
VAR4 --> hidden neuron 8	0.7886
VAR2(0) --> hidden neuron 8	2.7280
VAR2(1) --> hidden neuron 8	-4.2931
VAR2(2) --> hidden neuron 8	5.4967
VAR2(3) --> hidden neuron 8	-0.0529
VAR2(4) --> hidden neuron 8	-0.1573
VAR3(0) --> hidden neuron 8	1.8953
VAR3(1) --> hidden neuron 8	-0.0699
VAR3(2) --> hidden neuron 8	3.3665
VAR3(3) --> hidden neuron 8	0.4499
VAR3(4) --> hidden neuron 8	-1.8897
VAR5 --> hidden neuron 9	-2.7207
VAR6 --> hidden neuron 9	4.6578
VAR7 --> hidden neuron 9	-5.8605
VAR10 --> hidden neuron 9	1.0525
VAR9 --> hidden neuron 9	0.6142
VAR8 --> hidden neuron 9	3.1565
VAR12 --> hidden neuron 9	4.6107
VAR13 --> hidden neuron 9	-0.9566
VAR1 --> hidden neuron 9	7.8674
VAR11 --> hidden neuron 9	-0.1744
VAR4 --> hidden neuron 9	-0.1744

VAR2(0) --> hidden neuron 9	9.9895
VAR2(1) --> hidden neuron 9	-2.4867
VAR2(2) --> hidden neuron 9	1.8767
VAR2(3) --> hidden neuron 9	-3.9980
VAR2(4) --> hidden neuron 9	-1.9622
VAR3(0) --> hidden neuron 9	-3.1203
VAR3(1) --> hidden neuron 9	2.2926
VAR3(2) --> hidden neuron 9	0.0593
VAR3(3) --> hidden neuron 9	2.1369
VAR3(4) --> hidden neuron 9	2.0820
input bias --> hidden neuron 1	-1.5028
input bias --> hidden neuron 2	1.8353
input bias --> hidden neuron 3	4.4460
input bias --> hidden neuron 4	-1.2573
input bias --> hidden neuron 5	-16.0469
input bias --> hidden neuron 6	-4.7499
input bias --> hidden neuron 7	-3.8918
input bias --> hidden neuron 8	3.7264
input bias --> hidden neuron 9	3.4399
hidden neuron 1 --> collegeenroll(0)	-1.0605
hidden neuron 2 --> collegeenroll(0)	-1.2609
hidden neuron 3 --> collegeenroll(0)	1.5916
hidden neuron 4 --> collegeenroll(0)	-1.2746
hidden neuron 5 --> collegeenroll(0)	2.3928
hidden neuron 6 --> collegeenroll(0)	-1.1643
hidden neuron 7 --> collegeenroll(0)	-0.6351
hidden neuron 8 --> collegeenroll(0)	0.5634
hidden neuron 9 --> collegeenroll(0)	2.0249
hidden neuron 1 --> collegeenroll(1)	1.8399
hidden neuron 2 --> collegeenroll(1)	1.2402
hidden neuron 3 --> collegeenroll(1)	-2.0504
hidden neuron 4 --> collegeenroll(1)	1.3873
hidden neuron 5 --> collegeenroll(1)	-3.3982
hidden neuron 6 --> collegeenroll(1)	2.2255
hidden neuron 7 --> collegeenroll(1)	1.5395
hidden neuron 8 --> collegeenroll(1)	-1.2260
hidden neuron 9 --> collegeenroll(1)	-1.8381
hidden bias --> collegeenroll(0)	-1.3377
hidden bias --> collegeenroll(1)	-0.7987
VAR5 --> hidden neuron 1	-2.4271
VAR6 --> hidden neuron 1	-10.2395
VAR7 --> hidden neuron 1	3.1569
VAR10 --> hidden neuron 1	-1.2375
VAR9 --> hidden neuron 1	-0.4375
VAR8 --> hidden neuron 1	-0.5695
VAR12 --> hidden neuron 1	-6.4106

VAR13 --> hidden neuron 1	11.8356
VAR1 --> hidden neuron 1	-1.6717
VAR11 --> hidden neuron 1	-3.2228
VAR4 --> hidden neuron 1	-1.7481
VAR2(0) --> hidden neuron 1	0.7098
VAR2(1) --> hidden neuron 1	0.8935
VAR2(2) --> hidden neuron 1	7.0309
VAR2(3) --> hidden neuron 1	-6.8142
VAR2(4) --> hidden neuron 1	-3.3450
VAR3(0) --> hidden neuron 1	-5.0688
VAR3(1) --> hidden neuron 1	4.2943
VAR3(2) --> hidden neuron 1	-3.3745
VAR3(3) --> hidden neuron 1	2.6952
VAR3(4) --> hidden neuron 1	-0.0490
VAR5 --> hidden neuron 2	-2.7425
VAR6 --> hidden neuron 2	-0.7160
VAR7 --> hidden neuron 2	6.6629
VAR10 --> hidden neuron 2	4.1105
VAR9 --> hidden neuron 2	1.1869
VAR8 --> hidden neuron 2	-1.3314
VAR12 --> hidden neuron 2	-0.5877
VAR13 --> hidden neuron 2	0.7232
VAR1 --> hidden neuron 2	-4.1035
VAR11 --> hidden neuron 2	1.7972
VAR4 --> hidden neuron 2	-0.3031
VAR2(0) --> hidden neuron 2	3.1302
VAR2(1) --> hidden neuron 2	-3.9807
VAR2(2) --> hidden neuron 2	3.3228
VAR2(3) --> hidden neuron 2	-2.2593
VAR2(4) --> hidden neuron 2	1.6530
VAR3(0) --> hidden neuron 2	3.2038
VAR3(1) --> hidden neuron 2	3.2815
VAR3(2) --> hidden neuron 2	-5.5505
VAR3(3) --> hidden neuron 2	0.5486
VAR3(4) --> hidden neuron 2	0.2870
VAR5 --> hidden neuron 3	-3.1645
VAR6 --> hidden neuron 3	2.0882
VAR7 --> hidden neuron 3	-2.0796
VAR10 --> hidden neuron 3	-1.1709
VAR9 --> hidden neuron 3	0.7906
VAR8 --> hidden neuron 3	1.1038
VAR12 --> hidden neuron 3	-0.0380
VAR13 --> hidden neuron 3	1.1824
VAR1 --> hidden neuron 3	1.5225
VAR11 --> hidden neuron 3	-3.9238
VAR4 --> hidden neuron 3	1.2052

VAR2(0) --> hidden neuron 3	2.6315
VAR2(1) --> hidden neuron 3	-0.1310
VAR2(2) --> hidden neuron 3	0.3420
VAR2(3) --> hidden neuron 3	0.2160
VAR2(4) --> hidden neuron 3	1.3613
VAR3(0) --> hidden neuron 3	1.6898
VAR3(1) --> hidden neuron 3	3.8221
VAR3(2) --> hidden neuron 3	-3.2575
VAR3(3) --> hidden neuron 3	3.2081
VAR3(4) --> hidden neuron 3	-0.9738
VAR5 --> hidden neuron 4	4.3464
VAR6 --> hidden neuron 4	7.6987
VAR7 --> hidden neuron 4	-4.9896
VAR10 --> hidden neuron 4	-4.2760
VAR9 --> hidden neuron 4	6.9124
VAR8 --> hidden neuron 4	-2.8507
VAR12 --> hidden neuron 4	7.4019
VAR13 --> hidden neuron 4	9.3004

APPENDIX 6: Model ROC AUC Estimates by Grade

Table 37: Model ROC AUC Estimates by Grade

Model	Grade Level	ROC AUC Estimate
Artificial Neural Network	9	.8688
	10	.8784
	11	.8870
	12	.8926
Gradient Boosted Decision Trees	9	.8029
	10	.8378
	11	.8713
	12	.9108
Multilevel Logistic Regression	9	.8654
	10	.8780
	11	.8793
	12	.8986