Journal of Research High School

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

Predictive Modeling of College Enrollment: Harnessing Machine Learning to Forecast Student Educational Trajectories

Abhinav Garg[1] *

[1]Amador Valley High School, Pleasanton, CA, USA
*Corresponding Author: garg.abhinav0627@gmail.com

Advisor: Dr. Guillermo Goldsztein, ggoldsztein@yahoo.com

Received February 10, 2024; Revised May 19, 2024; Accepted, May 30, 2024

Abstract

In this paper, the primary objective is to create and validate a predictive model using neural networks to forecast high school students' likelihood of pursuing higher education to contribute to the body of knowledge in educational planning and policymaking. The foundation of the analysis rests upon a comprehensive dataset, encompassing diverse information about 1000 students. Pertinent factors considered include the nature of the educational institution attended, the institutional quality, gender demographics, and other relevant parameters. This study leverages the robust capabilities of Keras, a widely acclaimed open-source library nested within the TensorFlow framework. The modeling approach adopts a neural network architecture, featuring a sigmoid activation function in the output layer. To mitigate the potential risk of overfitting, this study integrates regularization techniques into the model construction process. The dataset undergoes a partitioning into a training dataset, constituting 75% of the samples, and a validation dataset, comprising the remaining 25%. The training process involves the application of the neural network on the training set, facilitating the refinement of the model's parameters. Subsequently, the validation set is employed to assess the model's generalization performance, affirming its efficacy in extrapolating insights to novel examples. This research not only showcases the utilization of cutting-edge machine learning tools but also emphasizes the significance of thoughtful data preprocessing and model validation methodologies. The results gleaned from this study contribute valuable insights into the predictive factors influencing a student's likelihood of pursuing higher education, thereby fostering a nuanced understanding of educational trajectories.

Keywords: Predictive modeling, Keras, TensorFlow, Neural networks, Regularization, Machine learning tools, Educational trajectories

1. Introduction

Machine learning, also known as artificial intelligence, is a field of computer science that uses data to make predictions and decisions (Alpaydin 2010). Machine learning has found applications in numerous fields. Examples include applications in the medical field, where machine learning is used for the diagnosis of diseases, such as heart disease, diabetes and pneumonia; applications in the banking business, where machine learning is used to make decisions on loan applications; applications to the real estate business, where machine learning is used to price real estate; applications to self-driving cars, where machine learning is at the core of the software used by self-driving cars; applications to machines' playing chess; and robots that can carry out numerous tasks (Mohri et al., 2018).

The task of predicting college enrollment is crucial for educational institutions, policymakers, and society at large, as it directly influences the strategic allocation of resources, the implementation of targeted interventions, and the overall accessibility of higher education. Research has shown that early identification of students who are less likely to pursue higher education can enable targeted support, thereby increasing enrollment rates and reducing educational

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

disparities (Bastedo & Jaquette, 2011). Furthermore, accurate predictions of college enrollment trends help institutions optimize their offerings and resources in response to expected demand, thus improving educational outcomes and student success (Hoxby & Avery, 2013). By employing predictive analytics, educators and policymakers can also better understand the multifaceted factors influencing students' decisions to pursue further education, including socio-economic backgrounds, academic performance, and personal aspirations (Domina et al., 2017). This understanding is pivotal in crafting policies and interventions that aim to make higher education more equitable and accessible to all segments of society, aligning with the goals of promoting lifelong learning opportunities and supporting economic development through a well-educated workforce (OECD, 2020).

Recent advancements in machine learning have significantly contributed to understanding of predictive modeling in educational settings. For example, Basu et al. (2019) utilized various supervised machine learning techniques to analyze data from over 11,000 students, aiming to predict college commitment decisions. Their study represented an essential step forward, leveraging logistic regression to achieve notable success in forecasting student decisions following admission offers. This research underscored the potential of machine learning to refine the predictions about student behavior, crucial for educational planning and resource allocation. Aulck et al. (2017) used numerous machine learning techniques including logistic regression, random forests, and k-nearest neighbors to predict student dropout in higher education. Moreover, a study by Golden et al. (2021) compared many different machine learning techniques used to predict a student's chances of admission to any university, underscoring the methodological diversity and numerous attempts made to develop predictive models in educational contexts.

However, despite these advancements, there remain considerable gaps in society's comprehensive understanding of college enrollment predictions. One notable limitation of existing research, including the study by Basu et al., is the focus on predicting immediate post-admission behaviors rather than addressing the broader, more complex question of what influences a student's decision to pursue higher education from a holistic perspective. Furthermore, much of the current literature has concentrated on singular aspects of the enrollment process, such as commitment post-admission or specific factors like academic performance and financial aid, without integrating these elements into a unified predictive model that encompasses a wider range of academic, socio-economic, and personal factors that influence a student's educational trajectory. Evidently, the lack of a predictive model to particularly predict a student's continuation into higher education leaves a hole in the understanding of forecasting students' educational trajectories.

In response, the primary objective of this study is to develop and validate a machine learning model capable of accurately predicting a high school student's likelihood of enrolling in higher education. This research seeks to bridge the gap in current academic literature by leveraging advanced neural network techniques, particularly focusing on the efficacy of using socio-economic and academic variables to forecast educational trajectories. By employing a comprehensive dataset and a rigorous methodological framework, this paper aims to provide insights into the key factors influencing college enrollment decisions. This study hypothesizes that a predictive model, developed using sophisticated machine learning techniques, will significantly outperform traditional statistical methods and simple regression models in forecasting college enrollment, demonstrating superior accuracy and reliability. The anticipated findings aim to contribute to the body of knowledge in educational planning and policymaking, offering a predictive tool that can assist in the development of targeted interventions to support student transitions to higher education. Moreover, the rigorous methodological approach adopted in this study ensures the robustness and validity of potential results, bolstering confidence in their applicability across different contexts. This confidence is vital, as it gives this research the potential to shape educational strategies and policies to enhance accessibility and equity in higher education. The computational model is built using a data set obtained from the website Kaggle (Mukti 2022). This is a website that has a large collection of data sets, available to the public, that can be used to develop machine learning models.

This paper is organized as follows. It is first explained what supervised learning is. This is a subclass of problems within the larger class of problems of machine learning. The predicting college enrollment example belongs to this category of supervised learning. This study then explains the structure of the data sets in supervised learning problems, and explain the concept of examples, features and labels, as well as the process known as integer encoding. This study explains these concepts in general, as well as in the going to college data set. Next, this study explains what logistic

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

regression models are; while these models are not used in the problem, it is instructive to start by explaining them. Then, this study explains what neural network models are. This is the class of models used to predict whether a student will go to college. This study explains the notion of parameters, training set, error on a set of examples, and how the parameters are selected by minimizing the error on the training set. This study also explains the technique of regularization and its use in addressing the issue of overfitting. This study finishes by illustrating the concepts explained by developing a model to predict whether a student will go to college. This study discusses the accuracy of the model on a set of examples that are not part of the training set. This set is called the validation set. The paper is finished with a small discussion and conclusion.

2. Supervised learning and the data set

The specific problem addressed is the prediction of college enrollment. This study utilizes a comprehensive synthetic dataset which closely resembles real data (due to the lack of available real data and data privacy concerns) derived from the machine learning website Kaggle, containing detailed records for 1,000 high school students aiming to predict their likelihood of enrolling in higher education. All features named come from research papers. Data correlation was measured using a correlation matrix to mimic real data before assigning labels (Mukti 2022). This dataset is publicly accessible for replication purposes and was used in accordance with Kaggle's terms of service. The information about each student is: the type of school the student attends, the quality of the school (where A is better than B), the gender of the student, the student's interest in going to college, the student's type of residence, their parent's age, their parent's monthly salary in IDR, their parent's house area in square meters, the student's average of grades, whether their parent was ever in college, and whether or not the student will go to college. These features are used so that academic and socioeconomic variables are accounted for. The selection of both academic and socioeconomic variables is grounded in existing literature (U.S. Department of Education, n.d.) indicating their significant influence on educational outcomes, thereby ensuring a comprehensive analysis. An important aspect of the dataset was its balance. The number of students who enrolled in college ('True') compared to those who did not ('False') was evaluated to determine if a class imbalance existed. With 500 instances for 'True' and 500 for 'False' within the dataset, the dataset is balanced. The balance of the dataset is a crucial factor in the development and evaluation of the predictive model. Imbalances can lead to misleadingly high accuracy, as models might predict the majority class more often. Therefore, ensuring the dataset is balanced is important for maintaining the integrity of the evaluation metrics. To prepare the data for machine learning analysis, integer encoding is applied to these categorical variables, where each unique category is assigned a unique integer value. Part of this data set is illustrated in Table 1. The entries in the first row are abbreviations of the information in each column. They have the following meaning:

The entry in the column TOS has a 1 if the student attends an academic school and a 0 if the student attends a vocational school.

The entry in the column QOS has a 1 if the student attends a school accredited as "A" and a 0 if the student attends a school accredited as "B."

The entry in the column GEN has a 1 if the student is male and a 0 if the student is female.

The entry in the column INT has a 4 if the student is very interested in going to college, a 3 if they are interested, a 2 if they are uncertain, a 1 if they are less interested, and a 0 if they are not interested.

The entry in the column RES has a 1 if the student lives in an urban residence and 0 if they live in a rural residence.

The entry in the column Age has the age of the student's parent.

The entry in the column SAL has the monthly salary of the student's parent in IDR, or Indonesian rupiah.

The entry in the column HA has the parent's house area in square meters.

The entry in the column GA has the student's average of grades on a scale from 0-100.

The entry in the column WIC has a 1 if the parent was ever in college and a 0 if the parent was not in college.

The entry in the column GTC has a 1 if the student will go to college and a 0 if the student will not go to college.

Table 1 shows the information about only two students, but the dataset contains information about 1000 students.

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

Table 1. Data of two of the examples in the data set.

TOS	QOS	GEN	INT	RES	Age	SAL	HA	GA	WIC	GTC
1	1	1	1	1	56	6950000	83.0	84.09	0	1
1	1	0	4	1	57	5250000	75.1	86.79	0	0

The problem considered in this paper belongs to the class of problems known as supervised learning. A first characteristic of this class of problems is that the data set consists of information about a collection of units. In the data set, the units are the students. In the language of machine learning, the units are called examples. Thus, the examples are the students in the data set.

A second characteristic about supervised learning problems is that the information the data set contains about each example is of two types: the label or target variable, and the features. The label is what one eventually wants to predict for examples that are not in the data set. In the data set considered, the label is whether the student will go to college or not. The rest of the information about each example is the features. Thus, in the data set, the features are the information stored in the columns TOS, QOS, GEN, INT, RES, AGE, SAL, HA, GA, and WIC.

Before these features are entered into the model, these features must be scaled. Feature scaling is the process of normalizing the data. Let $x_1, x_2, \ldots, x_n$ be a list of the features. The scaled list is then computed as $\frac{x_1-\mu}{\sigma}$, $\frac{x_2-\mu}{\sigma}$, …, $\frac{x_n-\mu}{\sigma}$ where μ is the mean of the features and σ is their standard deviation. Feature scaling helps ensure data is on the same scale and minimizes the effect of too large or too small values, thus helping the model's accuracy and performance.

The objective of the rest of this paper is to use the data set of the students to develop a computational model that can predict if a new student, not in the data set used to develop the model, will go to college. To make its prediction, one needs to provide the model with the features of the student. The rest of the paper will explain the theory behind the development of the model as well as the results obtained.

3. Binary Classification Problem

Each student will either go to college or not. In other words, the label takes one of two values: 1 if the student will go to college and 0 otherwise. Problems where the label takes one of two possible values are known as binary classification problems. Each example belongs to one of two categories, according to the value of its label. One of the categories is identified with the number 0 and the other with the number 1. The categories are called category 0 and category 1, respectively. In this case, 1 means the student will go to college and 0 means the student will not go to college.

A model for binary classification problems is a function that takes as input the features of an example and gives as output a number between 0 and 1. As is the common practice, this number is denoted by $\hat{y}$. As it will be explained soon, $\hat{y}$ is a prediction of the label of the example. Note that $\hat{y}$ is a function of the features of the example. In this case, each example has 10 features. These features are denoted by $x_1, x_2, \ldots, x_{10}$ and the meaning of the features are as in the columns of Table 1. Thus, given a student, $x_1 = 1$ if the student attends an academic school, but $x_1 = 0$ if the student attends a vocational school. Similarly, $x_2 = 1$ if the school's quality is A, but $x_2 = 0$ if the school's quality is B. The meaning of the other features, $x_3, \ldots, x_{10}$, is explained similarly from Table 1. Since $\hat{y}$ is a function of the features, written as $\hat{y} = \hat{y}(x_1, x_2, \ldots, x_{10})$. The prediction of the model is that the example with features $x_1, x_2, \ldots, x_{10}$ belongs to the category 1 if $\hat{y}(x_1, x_2, \ldots, x_{10}) > 0.5$ or to the category 0 if $\hat{y}(x_1, x_2, \ldots, x_{10}) < 0.5$.

It has not yet been explained how the function $\hat{y}(x_1, x_2, \ldots, x_{10})$ is selected. This will be done in subsequent sections. For now, consider the following example. Assume that a student has the following features:

$x_1 = 1$ (the student attends an academic school)

$x_2 = 1$ (the school's quality is A)

$x_3 = 0$ (the student is female)

$x_4 = 3$ (the student is interested in going to college)

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

$x_5 = 1$ (the student's residence is urban)
$x_6 = 47$ (their parent is 48 years old)
$x_7 = 10000000$ (their parent's monthly salary is 10 million rupiah)
$x_8 = 83.0$ (their parent's house area is 83 square meters)
$x_9 = 91.6$ (the student's average of grades is 91.6)
$x_{10} = 1$ (their parent went to college)
Assume that when these features are fed to the model, the output is 0.8, i.e.

$$\hat{y}(1,1,0,3,1,47,10000000,83.0,91.6,1) = 0.8.$$

This means that the model predicts that the student will go to college. The next sections describe how the function $\hat{y}(x_1, x_2, \ldots, x_{10})$ is constructed.

4. Logistic regression

Logistic regression is a machine learning technique that is used to develop models in binary classification problems. While this is not the technique that used in this paper, it is instructive to explain it in this section. It first must be explained what the sigmoid function is.

The sigmoid function is the function

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

The graph of the sigmoid function is displayed in Figure 1.

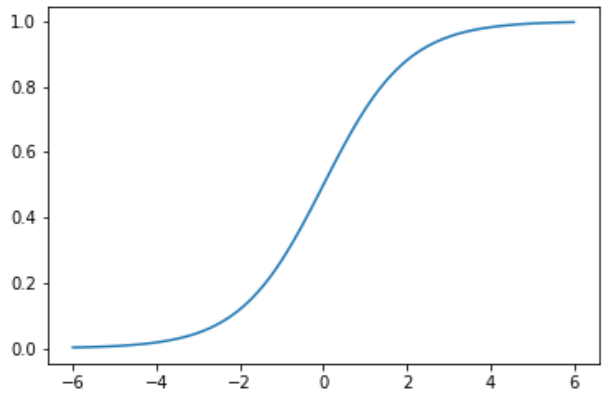

Figure 1. Plot of the graph of the sigmoid function.

The important properties of the sigmoid function are:

1. $0 < \sigma(x) < 1$ for all x.
2. $\sigma(x)$ is an increasing function of x.
3. $\sigma(x)$ becomes arbitrarily close to 0 as x becomes large in absolute value but negative.
4. $\sigma(x)$ becomes arbitrarily close to 1 as x increases.
5. $\sigma(0) = 0.5$.

In the case of the going to college problem, each example has 10 features. Logistic regression is a machine learning technique that assumes the prediction of the label to have the functional form

$$\hat{y} = \hat{y}(x_1, x_2, \ldots, x_{10}) = \sigma(w_1x_1 + w_2x_2 + \ldots + w_{10}x_{10} + b)$$

where, as before, $x_1, x_2, \ldots, x_{10}$ are the features of the examples, but $w_1, w_2, \ldots, w_{10}, b$ are some numbers known as parameters. The model is determined by the parameters. If the parameters are changed, the model changes and thus the predictions made by the model.

5. Neural networks

While very attractive for its simplicity, logistic regression has its limitations. If the data is not linearly separable, logistic regression will not work well. Fortunately, the ideas of logistic regression have been extended to methods that overcome the mentioned limitations. One of those methods, the one that used in this paper, is known as neural networks.

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

The function that used to make the predictions is a neural network. There are several types of neural networks. They will not be described in deep detail. Instead, the discussion will be focused on the basics of a fully connected neural network.

It is first discussed what is known as the architecture of a network, which is controlled by variables known as hyperparameters that one is free to select in the development of the model. The architecture can be described as follows:

1. A network is formed by a sequence of layers.
2. The first layer (layer 0) is called the input layer.
3. The last layer (layer L) is called the output layer.
4. The $L-1$ layers that are neither the input nor the output layer are called hidden layers.
5. Layer l has $n^{[l]}$ nodes. If one has k features and s labels, $n^{[0]}=k$ and $n^{[L]}=s$.
6. Each layer l has an associated activation function $f^{[l]}$.
7. In a binary classification problem, such as the problem of predicting whether a student will go to college or not, $f^{[L]}=\sigma$.
8. The input of the input layer is X, the features of the example (the student in this case).
9. The output of a layer is the input of the next layer
10. The output of the output layer is $\hat{y}$, the prediction of the label of the example by the network.

Figure 2 illustrates the architecture of a neural network.

There are several types of layers. The ones used in this paper are known as dense layers, which take as input a 1d-array with ℓ components and give as output a number.

Furthermore, there are several types of activation functions. The layers in the model except the output layer use the rectified linear unit (ReLU) function, which is defined as $f(x)=\max(0,x)$.

Figure 3 displays the graph of the ReLU function.

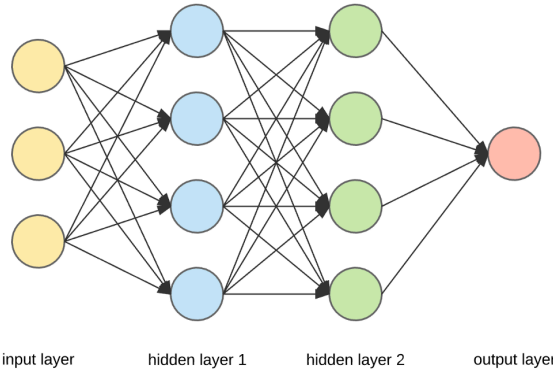

Figure 2. The architecture of a neural network.

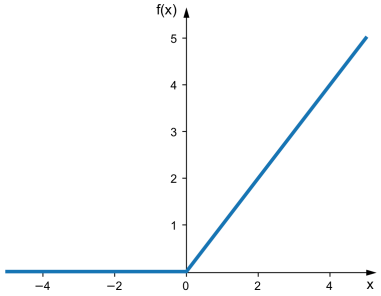

Figure 3. The graph of the ReLU function.

One reason this activation function is used in the model is due to its simplicity, as demonstrated in Figure 3. The model can thus take less time to train or run. Additionally, since the ReLU function gives output zero for all negative inputs, the network will be sparse, which results in a concise model with better predictions and less noise.

In addition, the network depends on a set of numbers known as parameters. As straight lines are determined by their y-intercept and slope, neural networks are determined by parameters. One should think of parameters in networks as the y-intercept and slope in straight lines.

For every node not in the input layer, there is a parameter. The parameter of node i in layer l is denoted $b_i^{[l]}$. Additionally, for every edge there is a parameter. If the edge connects node i of layer $l-1$ with node j of layer l, the parameter is denoted $w_{ij}^{[l]}$.

The rules that define this function are more complex than those of logistic regression, and are as follows:

1. Each node has data associated with it.
2. The data of node i of layer l is denoted by $a_i^{[l]}$.

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

3. The data of node i of layer 0 is x_i, the ith feature of the example. Thus, $a_i^{[0]} = x_i$.
4. For $1 \leq l \leq L$, $a_j^{[l]} = f^{[l]}\left(b_i^{[l]} + \sum_{i=1}^{n^{[l-1]}} a_i^{[l-1]} w_{ij}^{[l]}\right)$.
5. $\hat{y}_j = a_j^{[L]}$.

To create the model, one provides as input of the neural network the training set (which will be discussed in a later section) as well as the hyperparameters, and obtains as output the parameters $b_i^{[l]}$ and $w_{ij}^{[l]}$ which minimize the error on the training set.

The predictive model was constructed using Python, a versatile programming language favored for its extensive libraries supporting machine learning applications (Python.org, n.d.). Central to the model's development was the use of Keras, a high-level neural networks API, operating on top of TensorFlow, which provided the necessary infrastructure for designing and training deep learning models (Chollet et al., 2015). This study utilized the TensorFlow 2.x version, which offers an intuitive interface for constructing neural networks through layers of abstraction. Data preprocessing, essential for model accuracy, was conducted using the Pandas library for data manipulation and scikit-learn for scaling, ensuring the dataset was suitably formatted and normalized for training purposes. (McKinney et al., 2010; Pedregosa et al., 2011).

The neural network used in this study consists of an input layer with 10 nodes, corresponding to the number of features of a given example. The model has three hidden dense layers of 20, 8, and 2 nodes respectively. Each uses the ReLU activation function, which was explained earlier in this section. The output layer consists of a single node with the sigmoid activation function, since this is a binary classification problem, as described previously. The output of the model, $\hat{y}$, is a number between 0 and 1 where $\hat{y} > 0.5$ means the model predicts the student will go to college and $\hat{y} < 0.5$ means the model predicts the student will not go to college, as discussed earlier.

6. Binary cross entropy error

Assume that the features of an example are $x_1, x_2, \ldots x_{10}$. Assume that the label of that example is known and that this label is y. Note that y is either 1 or 0. On the other hand, the model predicts the label of this example to be $\hat{y}$. Note that $0 < \hat{y} < 1$. The binary cross entropy error on this example is defined to be $BCE(y, \hat{y}) = -(y \log(\hat{y} + (1 - y)\log(1 - \hat{y}))$.

While this paper will not go into the details of the binary cross entropy error, it is listed here its properties that are most relevant:

1. $BCE(y, \hat{y}) \geq 0$.
2. If $y = \hat{y}$, then $BCE(y, \hat{y}) = 0$.
3. The closer y is to $\hat{y}$, the smaller $BCE(y, \hat{y})$ is.

For the reasons listed above, $BCE(y, \hat{y})$ is a measure of the difference between y and $\hat{y}$. Thus, $BCE(y, \hat{y})$ can be considered as a measure of the error the model makes in predicting the label of the example. For example, assume that $y = 1$ and $\hat{y} = 0.7$, then

$$BCE(y, \hat{y}) = BCE(1,0.7) = -\log(0.7) = 0.15.$$

On the other hand, if $y = 1$ and $\hat{y} = 0.9$, then

$$BCE(y, \hat{y}) = BCE(1,0.9) = -\log(0.9) = 0.05.$$

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

Notice that the better prediction of $y = 0.9$ gave the smaller cross entropy error. Thus, the smaller the error, the better the predictions, and it is sought to minimize this error to obtain the best predictions.

The mean binary cross entropy error on a set of examples is the average of the binary cross entropy errors on the examples in the set. This is illustrated with the help of Table 2, where the labels y, the predicted labels y and the binary cross entropy errors $BCE(y, \hat{y})$ of three examples are displayed. The average of those errors is also shown, which is the mean binary cross entropy error on this set of three examples.

Table 2. Binary cross entropy errors of three examples and the mean binary cross entropy error on the set of these three examples together.

y	$\hat{y}$	$BCE(y, \hat{y})$
1	0.9	0.05
0	0.2	0.1
0	0.1	0.05
Mean $BCE(y, \hat{y})$		0.67

7. Training and validation sets

The examples on the data set used to develop the model are split into two sets: the set of training examples, or the training set, and the set of validation examples, or the validation set. As is common practice, the training set will contain 75% of the examples and thus, the validation set will contain 25% of the examples. This split is done randomly. In other words, given an example in the original data set, the probability that this example will belong to the training set after the split is 75%. Note that both the features and the labels of the examples are in both the training and the validation set. The reason for this split is described in later sections.

8. Selection of the parameters

Note that this binary cross entropy error on the training set depends not only on the values of the features and labels of the examples in the training set, but also on the parameters $w_{ij}^{[l]}$ and $b_i^{[l]}$. If those parameters are changed (keeping the training set the same), the binary cross entropy error also changes.

In a neural network the parameters that are selected are those that make the mean binary cross entropy error on the training set as small as possible. This paper will not go into any details on the algorithms used to find those parameters. In practice, these parameters are usually found using software libraries that are available to be used by the public at no cost. In this case, the Keras library is used to select the parameters that minimize the mean binary cross entropy error.

To illustrate the above discussion, consider Table 3, where a training set is shown with only six training examples. Each example has only one feature, so this Table is unrelated to the going to college data set considered in this paper, where each example has 10 features. In that table, MBCE means the mean binary cross entropy error. Note that, with the parameters $w = 1$ and $b = 0$, the mean binary cross entropy error is 4.56. On the other hand, with the parameters $w = 3.83$ and $b = -0.89$, the mean binary cross entropy error is 0.33. This means that the model with the parameters $w = 3.83$ and $b = -0.89$ is better than the model with the parameters $w = 1$ and $b = 0$. This is evident by looking at the column with the predictions y from each model. In fact, the parameters $w = 3.83$ and $b = -0.89$ gives the smallest mean binary cross entropy error, i.e. a model with other parameters gives a larger mean binary cross entropy error. Note also that this paper has not, and will not, explained how these optimal

Table 3. Example that illustrates that the parameters that lead to the smallest possible mean cross entropy error leads to better predictions.

x= feature	y= feature	$\hat{y}$= predicted label with $w = 1$ and $b = 0$	$\hat{y}$= predicted label with $w = 3.83$ and $b = -0.89$
-1	0	0.27	0.01
-0.8	0	0.31	0.02
0.2	1	0.55	0.47
0.4	0	0.60	0.66
0.8	1	0.69	0.90
1	1	0.73	0.95
MBCE		4.56	0.33

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

parameters, $w = 3.83$ and $b = -0.89$ are found. This paper only mentions that the Keras library is used to find these optimal parameters.

Now go back to the go to college data set. The corresponding training set is used to find the optimal parameters, i.e. the parameters that minimize the mean binary cross entropy error on the training set.

9. Overfitting and regularization

In most data, the value of the labels is not truly a function of the features. The value of the labels is made up of two components, the deterministic and the random components. The deterministic component is a function of the features, but the random component is not. Denote by y, y_d and y_r the label, the deterministic component of the label and the random component of the label, respectively. While y and y_r are functions of the examples, y_d is a function only of the features of the examples. In other words,

$$y(\text{example}) = y_d(\text{features of the example}) + y_r(\text{example})$$

Denote the features by x. The deterministic component is the average of the labels of all the examples whose features have the same values, or in other words,

$$y_d(x) = \text{average of } y \text{ over examples whose values of their features equal } x$$

To illustrate the concepts introduced above, consider as an example a dataset where the examples are persons, the features are the their weight, and their labels are their height. In this case, $y_d(x)$ is the average height of all the persons whose weight is x.

The goal of supervised learning is to come up with a model that predicts y_d, the deterministic component of the label. In the context of the example of the previous paragraph, the best one can hope for is to develop a model that given a weight x, predicts the average height of people with that weight.

Note that the random component of the label, y_r, is unique to the example under consideration. Thus, knowing y_r for an example does not help predict the label of another example. Going back to the weight-height context, y_r of a person is how much taller (if $y_r > 0$, shorter if $y_r < 0$) the person is in comparison to the average height among persons with the same weight.

Sometimes a model predicts the labels of the examples very closely, but it does not predict the deterministic component of the label that well. Consequently, the model does not work as well on examples that are not part of the training set. The model is being influenced too much by the random components of the labels of the training set. This effect is known as overfitting.

Note that if one continues increasing the complexity of the model, the error on the training set continues decreasing. But this is due to the model capturing the random components of the labels in the training set too much. This leads to overfitting. As a result, the error on the validation set eventually stops decreasing and starts increasing. This is an undesirable effect.

Because the model is quite complex, this paper considers several strategies to prevent or limit overfitting. The one used is known as regularization. Regularization is a pivotal method in machine learning that constrains or shrinks model coefficients, thereby preventing the model from fitting too closely to the training data and ensuring it generalizes well to unseen data. This paper adopts L2 regularization (Ridge) for its neural network model, given its effectiveness in reducing overfitting by penalizing large weights.

As before, the data is split into the training and validation sets, the reason for which is described in the next section. Let n be the number of examples in the training set. Then, $J_{\text{reg}} = J + \frac{\lambda}{n}\sum (w_{ij}^{[l]})^2$ where J is the binary cross entropy error. The parameters $w_{ij}^{[l]}$ and $b_i^{[l]}$ are selected such that they minimize J_{reg}. Note that λ is a hyperparameter. The model is trained several times with different values of λ, known as the regularization strength, and the model that gives the smallest error on the validation set is kept.

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

Observe that while minimizing J selects parameters that give good predictions on the training set, the addition of the second term in J_{reg} penalizes large values of the parameters $w_{ij}^{[l]}$. Keeping the size of these parameters small smooths the model. Thus, regularization enables both keeping the predictions on the training set good while keeping the size of the parameters in check.

Several different values of λ are tried and the results are listed in Table 4.

Table 4. Error on the validation set for different values of λ.

λ	J_{val} = error on the validation set
0	1.514
0.01	0.288
0.1	0.696
1	0.696
10	0.696

It is seen that $\lambda = 0.01$ minimizes the error on the validation set. Thus, this value of λ is selected to train the model, striking an effective balance between model complexity and predictive accuracy. This selection underscores the importance of regularization in enhancing the model's generalization capabilities.

Through the strategic application of L2 regularization, this study reinforces the model's robustness against overfitting. The regularization approach, coupled with hyperparameter optimization, underscores a methodical framework for ensuring the predictive model is not only accurate on the training data but also performs reliably on unseen data, thus enhancing the model's applicability and reliability in predicting college enrollment.

10. Validation set and evaluation

The validation set is used to evaluate how good the model is. The validation set was not used in the development of the model, thus, the validation set gives an accurate prediction of how well the model will work on new examples, these are examples where the label is not known. The model is evaluated on the validation set for this reason.

To assess the performance of the predictive model, a classification report was generated, providing a detailed analysis of the precision, recall, and F1-score for each class alongside overall accuracy. Because the dataset is balanced, as previously noted, simply measuring the overall accuracy is enough for determining the efficacy of the model; however, it is valuable to assess the other metrics.

Precision measures the model's ability to correctly identify positive instances for each class, indicating the proportion of true positives against all positive predictions. The model demonstrated precision scores of 0.86 for students not enrolling in college (labeled as 'False') and 0.90 for those enrolling (labeled as 'True'), reflecting its higher precision in identifying students who are likely to enroll.

Recall indicates the model's capability to identify all actual positive instances. Here, the model achieved recall scores of 0.89 for 'False' and 0.88 for 'True', showcasing a balanced sensitivity towards both potential enrollees and non-enrollees.

The F1-score is the harmonic mean of precision and recall, providing a single score that balances both the false positives and false negatives. The F1-scores for 'False' and 'True' were 0.87 and 0.89, respectively, indicating a robust overall performance of the model with a slight edge in favor of correctly identifying true positives.

The model's overall accuracy, which indicates the proportion of total correct predictions, was 0.88. This suggests that the model is expected to correctly predict the enrollment status of students 88% of the time. Furthermore, the macro average and weighted average scores for precision, recall, and F1-score were consistently 0.88. These averages corroborate the model's consistent performance across classes, considering both the balance and imbalance of the dataset.

The presented metrics collectively affirm the predictive model's efficacy, demonstrating its capability to serve as a reliable tool in forecasting college enrollment.

11. Discussion

This study has potential limitations. Firstly, the dataset utilized, derived from a singular geographic region, may not fully encapsulate the diversity of factors influencing college enrollment globally, potentially limiting the

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

generalizability of the findings. Additionally, the model's reliance on synthetic data introduces disadvantages, as it may not accurately reflect real-world complexities or current trends. Furthermore, the complexity of the neural network model, although beneficial for capturing intricate patterns in the data, may obscure the interpretability of which specific features most significantly influence predictions, hindering the direct application of findings for policy or intervention strategies. Addressing these limitations in future research could enhance the robustness and applicability of predictive models in the educational domain.

Nevertheless, this study's neural network model represents a significant advancement in predicting college enrollment, offering improved accuracy over traditional methods like logistic regression and decision trees. Logistic regression, popular for its simplicity and interpretability, often fails to capture complex variable interactions, limiting its predictive accuracy. Similarly, decision trees and their ensemble, random forests, although robust, are prone to overfitting when faced with high-dimensional data (Levy & O'Malley, 2020). The neural network model developed in this research not only addresses these limitations through advanced regularization techniques but also achieves a higher accuracy rate of 88%, compared to the 82.59-86.18% range reported by previous models and 56-79.22% for neural networks applied to comparable problems (Basu et al., 2019; Golden et al., 2021).

The model's ability to process a broad set of features and model nonlinear relationships without explicit feature engineering stands out as its primary advantage. However, the potential for further improvements exists, particularly through the exploration of more diverse datasets and alternative neural network architectures, such as recurrent neural networks for sequential data analysis.

The findings of this study underscore the potential of machine learning in transforming educational planning and policymaking. By accurately predicting college enrollment, this model equips educational institutions and policymakers with a powerful tool to identify students who may require additional support and resources to pursue higher education. Such predictive insights can lead to targeted interventions, thereby reducing educational disparities and promoting equal opportunities for all students. Moreover, this research highlights the critical role of data-driven approaches in understanding and addressing the multifaceted challenges facing the education sector today. It paves the way for future studies to explore more nuanced predictive factors and modeling techniques, further refining the accuracy and applicability of predictive models in educational settings.

The promising results of this study pave the way for several future research directions that can further enhance the predictive accuracy and applicability of machine learning models in education. Firstly, expanding the dataset to include a wider geographic and demographic scope could help in understanding regional differences and the impact of diverse socioeconomic factors on college enrollment. Additionally, incorporating longitudinal data could offer insights into how changes over time in individual students' circumstances or broader educational policies affect enrollment decisions. Exploring the integration of more complex machine learning algorithms, such as deep learning and ensemble methods, might also improve model performance by capturing more nuanced patterns in the data. Another valuable direction would be the development of interpretable models that provide not just predictions but also insights into the relative importance of different factors influencing college enrollment. This could aid educators and policymakers in designing more effective interventions. Finally, conducting similar studies in different educational contexts, such as vocational training or graduate school enrollment, would broaden the understanding of educational trajectories and guide tailored support strategies across the spectrum of learning pathways.

12. Conclusion

In the pursuit of understanding and enhancing the trajectory of students towards higher education, this paper presented a comprehensive exploration of machine learning techniques to predict college enrollment. Employing a dataset comprising a thousand students' academic and socio-economic backgrounds, a predictive model was developed using neural networks, facilitated by Keras and TensorFlow, to forecast the likelihood of students' continuation into higher education. The model's methodology was rigorously designed to address potential overfitting through regularization techniques, ensuring its generalization capabilities across unseen data. The validation of the model revealed a commendable predictive accuracy greater than the existing predictive models in the field, substantiating

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

the model's utility in identifying students who might require additional guidance and support to pursue further education.

The significance of this research extends beyond its technical achievements, embodying a crucial step towards the democratization of education. By pinpointing factors that influence college enrollment, the study not only aids educational institutions in tailoring their interventions but also assists policymakers in crafting policies that bridge the educational divide, ensuring that every student has the opportunity to realize their potential in higher education. Moreover, the study highlights the indispensable role of data-driven insights in navigating the complexities of educational outcomes, thus encouraging a more informed approach to educational planning and support mechanisms.

The implications of this research present numerous pathways for future investigation. Expanding the dataset to encapsulate a broader demographic and geographic spectrum, integrating longitudinal studies to capture the dynamic nature of educational pathways, and exploring advanced machine learning architectures are steps that can enhance the model's predictive power and applicability. Furthermore, the development of interpretable models could better unravel the factors influencing college enrollment, providing actionable insights for targeted interventions.

Acknowledgment

I would like to thank Dr. Guillermo Goldsztein for his mentorship and support completing this project. Without his guidance, this project would never have been completed.

References

Alpaydin, E. (2024, January 12). *Introduction to machine learning*. MIT Press. https://mitpress.mit.edu/9780262012430/introduction-to-machine-learning/

Archived: Factors Related to College Enrollment, www2. (2024).

Aulck, L., et al. (2017, March 7). *Predicting student dropout in higher education*. arXiv.org. https://arxiv.org/abs/1606.06364

Bastedo, M. N., & Jaquette, O. (2011). Running in place: Low-income students and the dynamics of higher education stratification. *Educational Evaluation and Policy Analysis*, 33(3), 318-339. DOI:10.3102/0162373711406718

Basu, K., et al. (2019). Predictive models of student college commitment decisions using machine learning. Data, 4(2), 65. doi:10.3390/data4020065

Chollet, F., et al. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Domina, T., et al., (2017). Is Free and Reduced-Price Lunch a Valid Measure of Educational Disadvantage? *Educational Researcher*, 46(7), 422-431.

Golden, P., et al. (2021). A comparative study on university admission predictions using Machine Learning Techniques. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 537–548. https://doi.org/10.32628/cseit2172107

Hoxby, C. M., & Avery, C. (2013). The missing "one-offs": The hidden supply of high-achieving, low-income students. *Brookings Papers on Economic Activity*, 2013(1), 1-65. https://www.brookings.edu/wp-content/uploads/2016/07/2013a_hoxby.pdf

Levy, J. J., & O'Malley, A. J. (2020). Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. BMC Medical Research Methodology, 20(1). doi:10.1186/s12874-020-01046-3

Vol. 2024 (8) 285 – 297
ISSN 2688-3651 [online]

McKinney, W., et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Mohri, M., Talwalkar, A., & Rostamizadeh, A. (2024, January 12). *Foundations of Machine Learning*. MIT Press. https://mitpress.mit.edu/9780262039406/foundations-of-machine-learning/

Mukti, S. S. J. (2022, June 29). *Go to college dataset*. Kaggle. https://www.kaggle.com/datasets/saddamazyazy/go-to-college-dataset/data?select=data.csv

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Müller, A. C., & Guido, S. (n.d.). *Introduction to machine learning with python*. O'Reilly Online Learning. https://www.oreilly.com/library/view/introduction-to-machine/9781449369880/

OECD. (2020). Education at a Glance 2020: OECD Indicators. OECD Publishing. https://doi.org/10.1787/69096873-en

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Python 3.0 release. Python.org. (n.d.). https://www.python.org/download/releases/3.0/