**Machine Learning Applied to Students Decisions on Going to College**

By: Ava Meier

Machine learning refers to a field of computer science that is used in the creation of models to make predictions and decisions. In this article, we apply machine learning to predict if students will attend college given their circumstances, which include the high school they attend, the salary of their parents and other environmental factors.

**1 Introduction**

Machine learning is a field of computer science that uses data to make predictions and decisions [1,2,3,4]. 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; machines experts in playing chess; robots that can carry out numerous tasks.

Machine learning has also found application in the social sciences. In this article, our goal is to predict if high school students will attend college. We want to make this prediction based on the students' life circumstances. These circumstances include factors such as the type of high school the student is attending and the salary of the student’s parents. Understanding the factors we consider here, may have applications in the decision of policies, such as the allocation of funding to support schools or improve social circumstances.

In this article, we use machine learning to develop a computational model to [predict if a student will attend college. The computational model is built using a data set that we obtained from the website Kaggle [5]. 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 article is organized as follows. We first explain what supervised learning is. This is a class of problems within the larger class of problems of machine learning. Our project belongs to this category of supervised learning. We explain the structure of the data sets in supervised learning problems, and explain the concept of examples, features and labels. We explain these concepts in general, as well as in our data set. Next, we explain what logistic regression models are. This is the class of models we use to predict if a student will attend college. We explain 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. We finish this article by illustrating the concepts explained by developing our model. We finish this project with a small discussion.

**2 Our Data Set**

Our data set consists of information about students. The information about each student is:

1. Type of high school that student attends
2. Quality of school
3. Gender of student
4. How interested are students if they go to college
5. Type of residence
6. Parent age
7. Parent salary
8. Parent house area in meter square
9. Average of grades
10. If a parent was ever in college
11. If the student attends college

Our dataset contains information about 1000 students.

**3 Supervised Learning**

The problem we consider in this article belongs to the class of problems known as supervised learning. The first characteristic of this class of problems is that the data set consists of information about a collection of units. In our 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 our 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 we eventually want to predict for examples that are not in our data set. In the data set we consider, the label is whether the student attends college or not. The rest of the information about each student is the features. Thus, in our data set, the features are: the type of high school that student attends, the quality of school, the gender of student, how interested are students if they go to college, the type of residence, the parent’s age, the parent’s salary, the parent’s house area in meter square, the student average of grades in high school, and if a parent was ever in college.

The objective is to use the data set of the students to develop a computational model that can predict if a new student will attend college or not. By a new student we mean a student not in the data set. To make this prediction, we will need to provide the model with the features of the student, but we won’t know the label. The goal is to predict the label, i.e., if the student will attend college. In the rest of the article, we will explain the theory behind the development of the model as well as the results we obtain.

**4 Binary Classification Problem**

Each student will either attend college or not. This means that the label of each example takes one of two values. One of the values is that the student will attend college and the other value is that the student will not attend college.

Problems where the label takes one of two possible values are known as binary classification problems. We say that 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. We call the categories category 0 and category 1, respectively. In our case, 1 means the student will attend college, and 0 means the student will not attend college.

In machine learning, 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, we denoted this number 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 our case, each example has 10 features. We denote these features by $x\_{1},x\_{2},...,x\_{10}$. Since $\hat{y}$ is a function of the features, we write $\hat{y}=\hat{y}(x\_{1},x\_{2},...,x\_{10})$. The prediction of the model is that the example with features $x\_{1},x\_{2},...,x\_{10}$ belongs to the category 1 if $\hat{y}(x\_{1},x\_{2},...,x\_{12})>0.5$ (this means that the prediction is that the student will attend college) or to the category 0 if $\hat{y}(x\_{1},x\_{2},...,x\_{12})<0.5$ (the student will not attend college).

We have not explained how the function $\hat{y}(x\_{1},x\_{2},...,x\_{10})$ is selected. We will do so in subsequent sections.

**5 Logistic regression**

Logistic regression is a machine learning technique that is used to develop models in binary classification problems. This is the technique that we use in this article and that we explain in this section. We first need to explain what the sigmoid function is.

The sigmoid function is the function

$$σ(x) = \frac{1}{1+e^{-x}}.$$

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

In the rest of this article, we assume that each example has 10 features, even when we talk in general terms, not just referring to our problem of predicting if a student will attend college. Logistic regression is a machine learning technique that assumes the prediction of the label to have the functional form

$$\hat{y}=\hat{y}(x\_{1},x\_{2},...,x\_{10})=σ(w\_{1}x\_{1}+w\_{2}x\_{2}+...+w\_{10}x\_{10}+b),$$

whereas before, $x\_{1},x\_{2},...,x\_{10}$ are the features of the examples, but $w\_{1},w\_{2},...,w\_{10},b$ are some numbers known as parameters. Note that we have not explained how the parameters are to be selected. We will get to that in subsequent sections. For now, note that the model is determined by the parameters. If we change the parameters, the model changes and thus, the predictions made by the model.

**6 Binary cross entropy error**

Assume that the features of an example are $x\_{1},x\_{2},...,x\_{10}$. Assume that we know the label of that example and this label is $y$. Note that $y$ is either 1 or 0. On the other hand, our 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

$$ BCE(y, \hat{y})= -(ylog( \hat{y})+(1-y)log(1- \hat{y}))$$

While we will not go into the details of the binary cross entropy error, we list here its properties that are most relevant to us:

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

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

We see that the better prediction of $ \hat{y}=0.9$ gave the smaller cross entropy error.

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. We illustrate this with the help of Table 2, where we display the labels $y$, the predicted labels $\hat{y}$ and the binary cross entropy errors $BCE(y, \hat{y})$ of three examples. We also show the average of those errors, which is the mean binary cross entropy error on this set of three examples.

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

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

**7 Selection of the parameters**

The data set given to us to develop the model, where both the features and the labels of each example are known, is called the training set

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 $w\_{1},w\_{2},...,w\_{10},b$. If we change those parameters (keeping the training set the same), the binary cross entropy error also changes.

In logistic regression, the parameters that are selected are those that make the mean binary cross entropy error on the training set as small as possible. We 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.

**8 Details of the Implementation**

In the rest of the article, we describe some details of the implementation

**8.1 Import Libraries**

Libraries are important for coding so that you don’t need to write your own code. The most important libraries I used during my project were NumPy (np) and pandas (pd). NumPy, which stands for Numerical Python, consists of multidimensional array objects and collection routines for processing arrays. Pandas, also known as panel data, has functions for analyzing, cleaning, exploring, and manipulating data.

**8.2 One Hot Encoding**

One hot encoding is a technique to represent categorical variables as numerical values in a machine learning model. I had to use this technique for seven columns in the dataset I started off with. Some of the categorical variables I changed were male and female, urban and rural, and true and false. To do this, I replace one of my categorical variables to the number one and the other to 0. If I had a column with more than two categorical variables like “Interest” in college, which had five, I changed it to numerical values that ranged from zero to four. By using this technique, I am one step closer to calculating the accuracy of the model.

**8.2 Prepare to Train the Model**

* 1. Assign the Output and the Input Variables

Setting variables to codes saves time in future coding. Instead of typing a full line of code, you can set the code equal to a variable. The variables I used were “X” and “y”.

* 1. Split into training and validation sets

A training set is what we use to create the model. A validation set is what we use to test how accurate the model is. I made the test size 0.25 and the random state equal to 4. The test size represents the portion of our test size, and the random state controls the random number generator used to shuffle the data before splitting it. It ensures that the same randomization is the same every time I run the code.

* 1. Scale the features of training and validation sets

By scaling the features of the training and validation set, it will make sure that all of the features will contribute equally to the result of determining how accurate this dataset is.

**8.3 Create the Model**

* 1. Train the model

The first code is to set the model equal to zero. This erases a model that I have created before, so it doesn’t affect any coding that I will do. Next, I set the model to sequential. This allows us to specify a neural network (teaches computers to process data similar to the human brain) precisely. After, I set the activation to both tanh and sigmoid, this shows us the range of the graph (see e.). Then, I compile the binary cross entropy. This increases the accuracy of my model. Finally, I set the epochs to 200 and the verbose to 0 and plot my model.

* 1. Visualize the model (plot and table)





* 1. Measure the quality of predictions

As you can see from the graphs above, the dataset that I started with has an accuracy of 89%. The dataset is 89% correct, so it means that when you use the model, you are going to be right 89% of the time.

**9 Conclusion**

When determining accuracy of datasets, python is a great tool to use to find an answer. I have learned to use machine learning to code and manipulate datasets and we see this through this article.

**References**

[1] Ethem Alpaydin. 2010. Introduction to Machine Learning (2nd ed.). MIT Press.

[2] Andreas C. Müller, Sarah Guido, 2016. Introduction to Machine Learning with Python. O'Reilly Media, Inc.

[3] Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar, 2018, Foundations of Machine Learning (2nd ed.). MIT Press.

[4] Kevin P. Murphy, 2012, Machine Learning A Probabilistic Perspective, MIT Press.

[5] <https://www.kaggle.com/datasets/saddamazyazy/go-to-college-dataset>