Assessing a Student's Interest in College Using Machine Learning

By Sampath Kalagarla

We reviewed machine learning concepts and applied them to a data set of students to create a model that identifies students who may go to college. The data set originally had 1000 students. After the removal of the students who either have parents who are too old or too young, the data set shrunk to 992 students. One of the most widely used open-source libraries, Keras, which is within TensorFlow, is used in this analysis. We use Neural Networks with a sigmoid activation function in the output layer. The dataset is divided into a training dataset (75%) and a validation dataset (25%). The model is trained on the training set. The validation set shows the model generalizes well to new examples.

Introduction to Machine Learning:

Machine Learning has become one of the most widely used methods for data analysis and forecasting recently [1-4]. It has been used in several industries to make critical decisions and optimize business processes such as stock trading, medical diagnosis, etc. Machine learning refers to a family of methods that use known information to develop computational models without direct instruction [1-4]. Machine learning is a subfield of artificial intelligence (AI). Machine learning uses algorithms to identify patterns in data, and uses those patterns to create a model that makes predictions. With increased data and experience, the results of Machine learning increase in accuracy, which is similar to how humans improve with more practice. Machine learning is particularly useful in scenarios where the data is always changing, the nature of the request or task is always shifting, or coding a solution would be effectively impossible [1-4].

Table 1: High School Student Dataset for the Decision to Attend College

type_school	school_accreditation	gender	interest	residence	parent_age	parent_salary	house_area	average_grades	parent_was_in_college	will_go_to_college
Academic	A	Male	Less Interested	Urban	56	6950000	83.0	84.09	False	True
Academic	A	Male	Less Interested	Urban	57	4410000	76.8	86.91	False	True
Academic	B	Female	Very Interested	Urban	50	6500000	80.6	87.43	False	True
Vocational	B	Male	Very Interested	Rural	49	6600000	78.2	82.12	True	True
Academic	A	Female	Very Interested	Urban	57	5250000	75.1	86.79	False	False

To illustrate our data set, we show in Table 1 the information about five students. The “True” or “False” in the last column refers to if a student will go to college. A student’s decision to attend college is impacted by various factors like their grades, residence, interest in attending college, and their parent’s salary. Table 1 lists 10 of the commonly considered factors.

One category of Machine learning problems is called supervised learning. The data in supervised learning problems consists of a collection of examples. In our case, the examples are the students. The information about each example is split into features and labels. The label here is whether the student is going to college or not, and the features are the rest of the information, i.e. “type_school”, “school_accreditation”, “gender”, “interest”, “residence”, “parent_age”, “parent_salary”, “house_area”, “average_grades”, and “parent_was_in_college”.

In this investigation, a High School dataset of 1000 students for college attendance is analyzed using Neural Networks and implemented with the software library Keras. The five students in Table 1 are examples of the dataset. The details of the analysis and the results are described in the rest of this article.

2. Data preprocessing

It is considered that attending college or not may have no significant correlation with the features, “parent_age”, “school_accreditation” and “school_tupe”. Thus, we remove these three features from the dataset. There are some parents much older or younger than the majority of parents, so it made sense to exclude 8 students from the data set. After doing this, we were left with 992 students with parents’ ages falling between forty-five and sixty years old.

In the column “gender”, we replace “male” by 0 and “female” by 1.

For each student, the “interest” feature has one of five possible values: “Less Interested”, “Very Interested”, “Interested”, “Uncertain”, and “Not Interested”. We apply one-hot encoding to this feature. More precisely, we replaced the feature “interest” with five new features: “Very Interested”, “Interested”, “Less Interested”, “Not Interested”, and “Uncertain”. If the “interest” of a student was “Less Interested”, they received a 1 in that feature and zeroes in the other four. The same applies for students with “interest” of “Uncertain”, “Not Interested”, “Interested”, and “Very Interested”.

After the steps described above, we were left with 494 students who are attending college and 478 students who are not attending. This is a fairly balanced set, so we don’t need to create any more students with the same features are other students already in the set.

Table 2 lists the information of the first five students after the preprocessing steps described in this section.

Table 2. High School dataset for Attending College after variable conversion and students with outlying information are removed.

type_school	school_accreditation	gender	residence	parent_age	parent_salary	house_area	average_grades	parent_was_in_college	will_go_to_college	Very Interested	Interested	Uncertain	Less Interested	Not Interested
Academic	A	0	Urban	56	6950000	83.0	84.09	False	True	0	0	0	1	0
Academic	A	0	Urban	57	4410000	76.8	86.91	False	True	0	0	0	1	0
Academic	B	1	Urban	50	6500000	80.6	87.43	False	True	1	0	0	0	0
Vocational	B	0	Rural	49	6600000	78.2	82.12	True	True	1	0	0	0	0
Academic	A	1	Urban	57	5250000	75.1	86.79	False	False	1	0	0	0	0

3. Training set and validation set

The set of examples is randomly split into two sets, the training set and the validation set. The training set contains 75% of the examples, and the validation set the rest 25%. Different computational models will be developed using only the training set. The validation set will be used only to evaluate the performance of the models. More details are given later in this article.

4. Binary classification problems

Data classification is a large domain in the field of statistics and machine learning. Generally, classification can be broken down into two areas: binary classification and multiclass

classification [1-4]. In binary classification, the labels of the examples take either the value 1 or the value 0. In our data set, 1 corresponds to the student going to college and 0 to the student not going to college. These two groups, i.e. the group of examples with label 1 and the group of examples with label 0, are called category 1 and category 0, respectively.

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 $\hat{y}$. The prediction of the model is that the example belongs to category 1 if $\hat{y} > 0.5$ and the category 0 if $\hat{y} < 0.5$.

5. Logistic regression

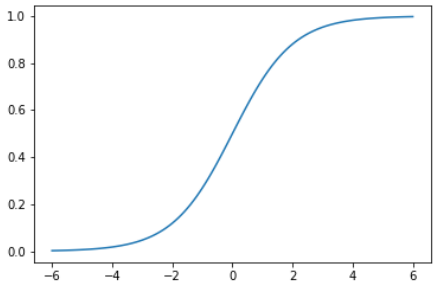
Logistic regression is a machine learning technique that is used to develop models in binary classification problems. We explain this technique in this section.

The sigmoid function is the function

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

The graph of this function is displayed in Figure 1.

Figure: Graph of the sigmoid function.



The properties of the sigmoid function that we will use are: For all values of x, we have that $0 < \sigma(x) < 1$. The sigmoid function is an increasing function. The value of $\sigma(x)$ approaches 1 as x increases. The value of (x) approaches 0 as x decreases. $\sigma(0) = 0.5$

We denote the features by $x_1, x_2, \ldots, x_{10}$ in the order they appear in Table 2. Thus, x_1 is the gender (0 or 1), x_2 the residence type (0 or 1), etc. In logistic regression, the functional form of the prediction ŷ is assumed to be of the form

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

where $w_1, w_2, \ldots, w_{10}, b$ are numbers known as parameters. To explain how these parameters are selected, we first need to introduce the notion of error.

6. Binary cross entropy error and selecting the parameters by minimizing the error

In this section we discuss how the parameters $w_1, w_2, \ldots, w_{10}, b$ introduced in the last section are selected.

Assume we know the label of an example and that this label is y. Note that y is either 1 or 0. On the other hand, let $\hat{y}$ be the prediction of our model of the label on this example. Note that $0 < \hat{y} < 1$. The binary cross entropy error in this example is

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

While we will not go into the details of the binary cross entropy error, the properties that are most relevant to us are: $BCE(\hat{y}, y) \geq 0$. The closer $\hat{y}$ is to y, the smaller $BCE(\hat{y}, y)$ is. If $\hat{y} = y$, then $BCE(\hat{y}, y) = 0$. Thus, $BCE(\hat{y}, y)$ is a measure of the difference between a $\hat{y}$ and y. In other words $BCE(\hat{y}, y)$ can be considered as a measure of the error the model makes in predicting the label of the example.

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. Note that this error depends on the parameters $w_1, w_2, \ldots, w_{10}, b$. These parameters that are selected are those that make 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 (such as Keras) that are available to be used by the public at no cost.

7. Measuring the quality of the model

The performance of a machine learning model is evaluated by the so-called "Accuracy" [5-7], which is the fraction of predictions the model got right. For a binary classification, the accuracy is calculated by dividing the total number of correct predictions, no matter positive or negative, by the total number of predictions. The accuracy varies from 0 to 1. The accuracy with a value close to 1 indicates that the model works well. In the opposite case where accuracy is close to 0 means the model has a poor performance. The accuracy may provide a good assessment of the performance of the model when the dataset is balanced, as is our case after the preprocessing of our data.

8. Applications

In this section, we apply the logistic regression and the neural networks approaches, described in previous sections, to analyze the data for students who may have attended college and to train the model to predict the possibilities of a student who may have attended college. Following are the parts of our codes for this analysis, from which the exact model, functions, and parameters used in this analysis are seen. Fig. 1 shows the cross entropy error versus the epoch (the number of iterations of model tuning).

```
model = Sequential()
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy')
model.fit(X_train_scaled,y_train,epochs=40,verbose=0)
J_list = model.history.history['loss']
plt.plot(J_list)
```

The binary cross entropy error on the training set is 0.3906903. The accuracy of the model on the training set is 0.83. After the model is trained with the training set, the model is applied to the validation set to validate the model. The binary cross entropy error on the validation set is 0.3660757, which is about the same as that on the training set. The accuracy on the validation set was found to be 0.84, very similar to what was obtained on the training set.

The model has also been trained with the training dataset at different epochs. With an epoch equal to 1000, the training error becomes 0.3313994, which is smaller than that when the epoch equals 40. The accuracy increases a little bit and becomes 0.93. The precision and recall are slightly higher than those for the epoch 40 as well. The model's performance is about the same when applied to the validation dataset. When the model is trained with an epoch of 2000, the cross-entropy error starts to increase. The accuracy becomes smaller compared when the number of epochs equals 2000 and the precision and recall become less consistent compared to the consistency between the two parameters when the epoch is limited to 1000. Thus, it is better to train the model with an epoch limited to 1000 or less.

Overfitting is a term that refers to a modeling error that occurs when a function corresponds too closely to a particular set of data. As a result, overfitting may fail to fit additional data, and this may affect the accuracy of predicting future observations. Overfitting can be identified by checking validation metrics such as accuracy and loss. If the model performs about the same on the validation dataset as on the training dataset, there is unlikely to be an overfitting issue in the model. In this investigation, our trained model performs about the same on the two datasets, the training dataset, and validation dataset. Therefore, there shouldn’t be any serious overfitting problem in the model trained in this data analysis for identifying potential students who may attend college.

Conclusion

A model based on machine learning concepts and libraries has been developed to predict if a student will attend college. The tools and the model “Sequential" in the two machine learning libraries, TensorFlow and Keras, are used to build the model. The model is trained and validated

with a dataset of students. The dataset is divided into a training dataset (75%) and a validation dataset (25%). The former one is used to train the model with logistic regression. The trained model is then applied to the validation dataset. The model works well on both the trained dataset and the validation dataset. The model performs well in both the training dataset and validation dataset with a similar accuracy of 0.7 when the epoch equals 40 and 0.76 when the epoch equals 2000. The dataset is not oversampled and is almost a 50-50 split between students attending college and those not attending college. The model will work well when it is applied to any new student to predict whether he will choose to attend college.

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. Nishant Shukla, 2018, Machine Learning with TensorFlow, Manning Publications Co.
6. GeÌron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly.
7. Giancarlo Zaccone, Md. Rezaul Karim, 2018, Deep Learning with TensorFlow, Packt Publishing Ltd.