Predicting Course Completions for Online Courses

Joseph Paetz
Final Paper
Machine Learning, CS 229
December 2014

Predicting Course Completions for Online Courses

Introduction

The Peace Operations Training Institute (POTI) is a small charity organization located in Williamsburg, Virginia that specializes in distributing peacekeeping related training courses to students located worldwide. They support approximately 20,000 students annually, providing anywhere from 60,000 to 100,000 course enrollments. The majority of their funding comes from grants provided by foreign nations and as a result POTI is able to provide the majority of their students their courses at no charge to the student. For grant reporting purposes, a number of metric goals are required including the need to measure how successful POTI is at training a student on the material presented in the course. If a student fails to complete the course (not to be confused with failing the course), it becomes that much more difficult to meet these goals.

The completion rate for courses that are distributed to students for free is low and it becomes a strong talking point for individuals that advocate for classroom style training vs online based training. Therefore it is important to find ways to raise the completion rate. Accurately identifying the individuals most likely to complete the courses and those individuals most likely to abandon the course becomes an important component of any strategy implemented to curb this problem.

The goal of my project is to determine if I can accurately predict if a student is going to complete an online course or abandon it. I will have access to all student data that the organization collects on registered students including basic demographic information (nationality, gender, organization) and their course history. The available data will encompass a total of approximately 300,000 course enrollments spanning the last 4 years.

Features

A total of 9 features were used to train the classifier:

1. Nationality: The student's nationality. The majority of the students training through POTI come from African nations. A possible point of concern is related to having too small of a sample from countries with low representation in the training data.

2. Gender: Indicates the student's gender.

3. Program ID: Registered students at POTI are placed into internal "programs" that indicate what pricing scheme the student will adhere to when enrolling in courses. An example program is "EDU" – indicating that a registered individual has a student status at a university.

4. Total course enrollments: This discrete numeric feature represents the total number of courses an individual student has signed up to study.

5. Completed previous enrollments: This will be a binary feature indicating if the student has ever completed a course in the past. They can either have failed or passed the course.

6. Total completions: This discrete numeric feature represents the total number of courses a student has completed.

7. Days between: This discrete numeric feature measures the number of days between the student's account creation date to the date the student signed up for the course.

8. Paid Money Previously: This binary feature indicates if a student has purchased anything from POTI in the past.

9. Completed: This is the value that the classifier is being trained to predict. It indicates if the course enrollment was completed by the student: to be considered a completed course, a final exam needs to be passed or failed 2 consecutive times.

Training Data Collection

The training data had to be extracted and transformed from the database tables. Categorical features were converted into equivalent 1/-1 values in order to be compatible with the SVM algorithm requirements:

Sample data prior to conversions

1	2	3	4	5	6	7	8	9
US	m	4	1	F	1	601	F	F
US	m	89	2	F	0	1039	F	F
US	m	89	2	F	0	1147	F	F
AU	m	89	1	F	0	1096	F	T
US	m	89	23	F	0	730	F	T
US	m	89	23	F	0	835	F	T
US	m	89	23	F	0	835	F	F
US	m	89	23	F	3	835	T	F
US	m	89	23	F	0	835	T	F
US	f	89	23	T	0	835	T	F

Sample data after conversions

1	2	3	4	5	6	7	8	9
187	1	4	1	-1	1	601	-1	-1
187	1	89	2	-1	0	1039	-1	-1
187	1	89	2	-1	0	1147	-1	-1
10	1	89	1	-1	0	1096	-1	1
187	1	89	23	-1	0	730	-1	1
187	1	89	23	-1	0	835	-1	1
187	1	89	23	-1	0	835	-1	-1
187	1	89	23	-1	3	835	1	-1
187	1	89	23	-1	0	835	1	-1
187	-1	89	23	1	0	835	1	-1

Algorithm Models

The following 3 algorithms will be used to build a classifier: SVM, Random Forest, and Nearest Neighbor. The following libraries written for the R programming language were used in conjunction with coding I wrote to create the classifier and to measure the misclassification error: **e1071** [SVM], **randomForest**, and **class** [nearest neighbor].

Sample R Code implementing Random Forest:

```
library(randomForest)
train<-read.csv("training.csv", sep="|",header=T)
test<-read.csv("testing.csv", sep="|",header=T)

train<-train[,3:11]
test<-test[,3:11]

y<-as.factor(train[,9])
x<-train[,1:8]

fit<-randomForest(x,y,ntree=250)
1-sum(y==predict(fit,x))/length(y)

y_test<-as.factor(test[,9])
x_test<-test[,1:8]
1-sum(y_test==predict(fit,x_test))/length(y_test)
```

Results

Using too large of a training set resulted in computer freezes, so I limited the training data to enrollments created in 2013. Testing data was related to enrollments created in the first 6 months of 2014.

Training Size: 62,850
Testing Size: 55,900

==SVM==
Kernel: Radial
Misclassification Error (Training): 7.868%
Misclassification Error (Testing): 7.765%

Kernel: Linear
Misclassification Error (Training): 8.99%
Misclassification Error (Testing): 9.289%

Kernel: Polynomial
Misclassification Error (Training): 7.886%%
Misclassification Error (Testing): 8.075%

Kernel: Sigmoid
Misclassification Error (Training): 15.215%
Misclassification Error (Testing): 12.64%

==Random Forest==
n-trees = 500
Misclassification Error (Training): 6.788%
Misclassification Error (Testing): 7.538%

==Nearest Neighbor==
too many ties in knn

I found that using an SVM model with a radial kernel resulted the best classifier for the training and test data. With less than an 8% classification error, I would state that we would be able to do a very good job at predicting if a student were going to complete the course or not at enrollment time.

Problems Encountered

There were 2 problems that were encountered when building a classifier using the algorithms listed previously. The Nearest Neighbor library used would return an error message stating that there were too many ties in knn. I was not able to come up with a combination of features that avoided this problem.

If I used a training set that was too large, I ended up crashing my computer. I ended up having to significantly reduce the size of my training data as a result.