Student Intervention System

Udacity Machine Learning Engineer
Nanodegree Program: Project 2

Anderson Daniel Trimm

February 5, 2016

1 Classification vs Regression

Our model predicts which students will pass or fail their high school final exam, and therefore need an intervention. Since we are predicting *discrete* labels (in this case, binary), this is a *classification* problem, as opposed to a *regression* problem, in which we would be predicting *continuous* labels.

2 Exploring the Data

In this section, we use the code in the accompanying ipython notebook to identify important characteristics of the dataset which will influence our prediction. Running the code, we find

```
Total number of students: 395
Number of students who passed: 265
Number of students who failed: 130
Number of features: 31
Graduation rate of the class: 0.67%
```

3 Training and Evaluating the Models

3.1 KNN Classifier

The k-nearest neighbor classifier finds the k training samples closest in distance to the query point, and predicts the label from these.

Pros:

- Fast training time - it simply remembers all the training points

- Being non-parametric, it can be useful in classification problems where the decision boundary is very irregular

Cons:

- Potentially long training time - it has to search through the dataset to find the k nearest neighbors
- Uses a lot of CPU memory, since it has to store the dataset (as opposed to a parametric model, which throws away the dataset after learning the parameters)

Despite the memory cost, since our dataset is not too large, kNN is a reasonable classifier to try.

3.2 Gaussian Naive Bayes Classifier

Naive Bayes is a learning algorithm based on applying Bayes' theorem with the assumption that every pair of features are independent (hence the "naive"). Here, we use the Gaussian naive Bayes classifier, which assumes the likelihood of each feature is Gaussian.

Pros:

-

Cons:

-

3.3 Support Vector Machine Classifier