Project Report: Student Intervention System

Aravind Battaje

March 19, 2016

1 Project Steps

TODO: Write later

2 Classification vs Regression

A machine learning algorithm can be classified into two types, based on its nature of outputs, viz., classification and regression. Classification supports outputs of discrete values and regression outputs continuous values. This project entails a classification type of problem because the output desired from the *intervention system* is discrete in nature, i.e., a student graduates or not from his/her current characteristics. Regression would be more suitable for, say an algorithm that predicts the final exam score from a student's current academic records.

3 Dataset

Several qualities of students such as their family background, social characteristics, extra-curricular activities, etc., along with the information if they graduated or not, are given along with the project `(student-data.csv)`. The dataset possesses following characteristics:

Total number of students	395
Number of students who passed	265
Number of students who failed	130
Graduation rate of the class	67.09%
Number of features of dataset	30

4 Training and Evaluating Models

Three supervised learning algorithms from `scikit-learn` were probed for their potential in *best* modeling the student intervention problem.

4.1 Naive Bayes Classifier

Naive Bayes Classifier is one of the simplest algorithms used in supervised learning.

	Training set size		
	100	200	300
Training time (msec)	1.136737	1.401565	1.677573
Prediction time - Training set (msec)	0.539596	0.743282	0.940869
Prediction time - Testing set (msec)	0.535090	0.538042	0.541995
F1 score - Training set	0.703436	0.800078	0.797350
F1 score - Testing set	0.613627	0.746451	0.752224

Table 1: Performance of Naive Bayes Classifier (100 runs)

	Training set size		
	100	200	300
Training time (msec)	1.436007	4.016979	8.038867
Prediction time - Training set (msec)	0.798676	2.626345	5.504694
Prediction time - Testing set (msec)	0.758820	1.313007	1.809123
F1 score - Training set	0.912564	0.903239	0.895141
F1 score - Testing set	0.794443	0.790969	0.792418

Table 2: Performance of SVC Polynomial 2^{nd} degree Kernel (100 runs)

	Training set size		
	100	200	300
Training time (msec)	1.720572	5.235305	10.758593
Prediction time - Training set (msec)	1.102505	3.869443	8.228962
Prediction time - Testing set (msec)	1.050613	1.893101	2.676311
F1 score - Training set	0.927031	0.911822	0.904793
F1 score - Testing set	0.800927	0.805108	0.808884

Table 3: Performance of SVC RBF Kernel (100 runs)

	Training set size		
	100	150	200
Training time (msec)	116.995811	116.348028	116.556168
Prediction time - Training set (msec)	8.880124	9.707942	10.348394
Prediction time - Testing set (msec)	8.139987	8.327084	8.233488
F1 score - Training set	0.951172	0.864298	0.819485
F1 score - Testing set	0.586345	0.614715	0.622481

Table 4: Performance of AdaBoost Classifier (100 runs)

4.2 Support Vector Machine

4.3 Boosting

5 Finding the Best Model

6 Notes

Use of Neural Networks was considered, but scikit-learn (stable) doesn't directly support multi-layer perceptron currently. Although other libraries (Theano, scikit-neuralnetwork) could be used, exploration has been

pushed forward both because it was suggested to "choose 3 supervised learning models that are available in scikit-learn", and neural networks will be later encountered during *Deep Learning*.