Assignment Report

Project 2: Building a Student Intervention System

Udacity Machine Learning Nanodegree Program
Authored by: Fedor Sulaev
Date: 02/06/2015

1. Classification vs Regression

The goal of the intervention system is to identify students who might drop out of school. The output of the system consists of 2 discrete groups of students. The system **classifies** students by the likelihood of them passing the exam. This is a classification problem it requires the system to label an input vector as a member of a class. Regression is used to predict real number or continuous output.

2. Exploring the Data

The dataset consists of 395 students, 265 students passed the exam and 130 failed. Graduation rate of the class is 67%. The dataset has 30 features excluding the label column.

3. Preparing the Data

The dataset is separated into feature and target columns. Non-numeric feature columns are converted to different format: binary columns that contain “yes” or “no” are converted to 1 and 0, other columns that have several possible values are split into multiple columns for each choice and assigned value 1 or 0. After preparation the dataset is shuffled and split into training and test sets.

4. Training and Evaluating Models

Three supervised learning models were tested on the dataset: Decision Tree, Gaussian Naive Bayes and Support Vector Machine.

4.1. Decision Tree

Decision tree model complexity is $O(n \log(n))$ for training and $O(\log(n))$ for prediction. Decision trees are helpful in situations of complex multistage decision problems. Advantages of decision trees are:

+ Simple to understand and to interpret.
+ Requires little data preparation.
+ The computational cost is logarithmic.
+ Able to handle both numerical and categorical data.
+ Able to handle multi-output problems

Disadvantages of decision trees are:

– Can create over-complex trees that do not generalize the data well (overfitting).
– Small variations in the data might result in a completely different tree being generated.
– Practical decision tree algorithms cannot guarantee to return the globally optimal decision tree.
– Decision tree learners create biased trees if some classes dominate.

Decision tree model might not be well suited for the given problem because of the small number of classes, that are not distributed evenly, on the other hand it is a simple algorithm with fast prediction and it can be tuned for better performance.

4.2.Gaussian Naive Bayes

Naive Bayes model complexity is $O(n)$ for training and prediction. Naive Bayes models are often used in text classification and spam filtering. Advantages of Naive Bayes classifiers are:

+ Require small amount of training data.
+ Fast training.
+ Each distribution can be independently estimated as a one dimensional distribution.
+ Easy to implement.

Disadvantages of Naive Bayes are:

− Bad at predicting probability.
− Cannot model dependencies between features.

Naive Bayes classifier model is very fast and universal algorithm that expected to perform well on the school dataset. It's main disadvantage in this case is that it can't learn dependencies between features.

4.3.Support Vector Machine

SVM model complexity can vary from $O(n^2)$ to $O(n^3)$ depending on the implementation. SVMs can be used to solve various real world problems such as text categorization, image classification, protein classification, hand-written character recognition etc. Advantages of SVM are:

+ Effective in high dimensional spaces.
+ Uses support vectors in the decision function.
+ Different kernel functions can be specified for the decision function.

Disadvantages of SVM are:

− If the number of features is much greater than the number of samples, the method is likely to give poor performance.
− Doesn't provide probability estimates.
− Slow algorithm.
− Black box, hard to interpret.

SVM is the most advanced model of the three, and expected to provide the best results. SVMs main disadvantage is high complexity which results in worse training and prediction time.

4.4.Model Comparison

Decision Tree			
	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.002
Prediction time (secs)	0.000	0.001	0.000
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.682	0.714	0.714

Gaussian Naive Bayes			
	Training set size		
	100	200	300
Training time (secs)	0.000	0.000	0.000
Prediction time (secs)	0.001	0.001	0.001
F1 score for training set	0.814	0.787	0.773
F1 score for test set	0.719	0.767	0.762

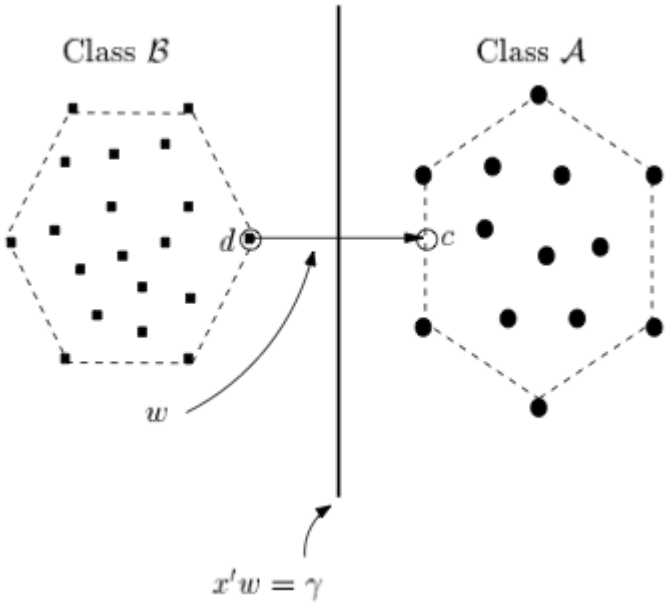
Support Vector Machine			
	Training set size		
	100	200	300
Training time (secs)	0.001	0.003	0.006
Prediction time (secs)	0.001	0.002	0.004
F1 score for training set	0.879	0.878	0.868
F1 score for test set	0.846	0.841	0.834

5. Choosing the Best Model

All three models were trained on training sets of different size and then tested on the same test set. Support Vector Machine model shows the best performance of 0.846 on training set containing 100 records. Gaussian Naive Bayes model has F1 score of 0.767 on training set with 200 records. Decision Tree model was the least accurate with F1 score of 0.714 on training sets with 200 and 300 students.

Gaussian Naive Bayes and Decision Tree models show better computational performance than SVM but the difference is negligible, worst case time for SVM is 0.004 seconds for prediction and 0.006 seconds for training, computation time for other models in all cases is under 0.002 seconds. However, complexity of SVM model in worst case is cubic and if the number of students is going to be significantly larger than the testing set, computational time might have to be taken into consideration.

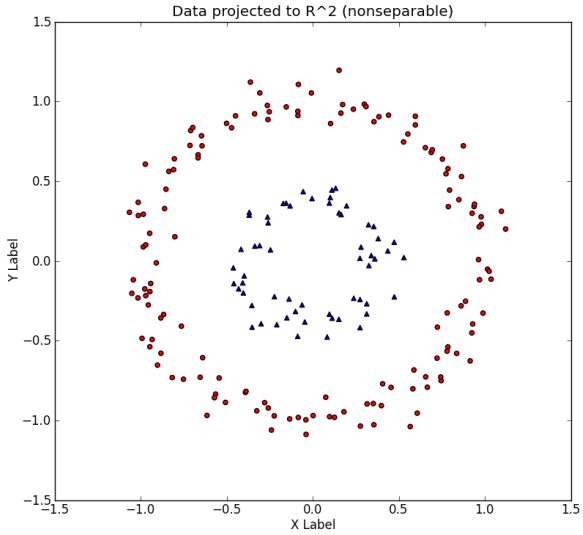
Support Vector Machine is the most appropriate supervised learning model for the student intervention system based on the test results and the available data. SVM purpose is to find the best separating line between points of different classes. Points in this case are entries in the dataset and can be represented graphically as points on a plane.

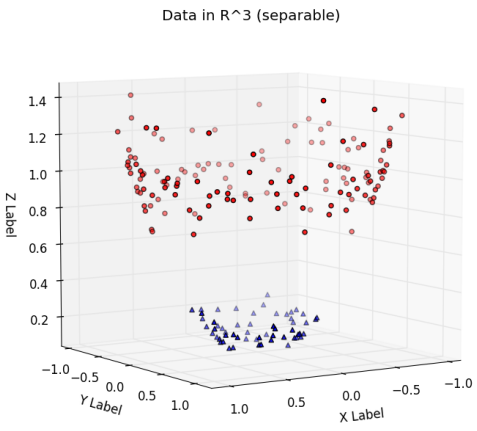


[1]

SVM searches for the closest points from different classes (support vectors) and draws a line connecting them (*w* on the picture). The SVM then declares the best separating line to be the line that bisects and is perpendicular to the connecting line. In this simplified example the data is separable by a straight line, but SVMs can be used for more complex datasets as well by using a "kernel trick".

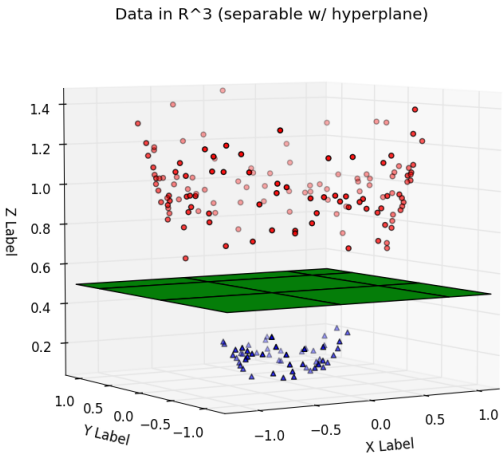
Kernel trick allows SVM to classify data points that cannot be separated by a straight line by mapping them to a space with more dimensions:

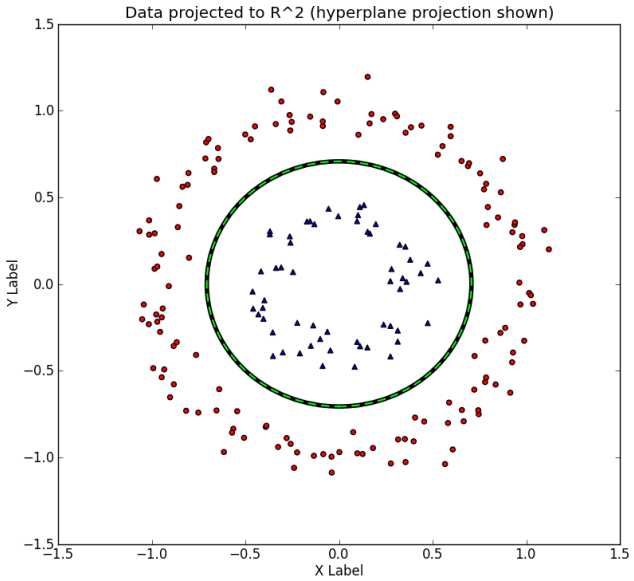




[1] "Duality and Geometry in SVM Classifiers" by Kristin P. Bennett and Erin J. Bredensteiner

The resulting separation plane is then projected back to 2D space:





SVM model was tested with different C and gamma parameters, no significant performance gain was observed from tuning the model. F1 score of the tuned model is approximately the same as for the default model.