Project: Build a Student Intervention System

Classification vs Regression

The task of identifying students who might need early intervention is a classification problem. In regression problems, we expect a quantitative output such as value of a variable. In classification problems, we separate the output or target into two or more classes. In this problem, the students can be divided into two groups, those who might need early intervention and those who do not. Classification machine learning algorithms can be used to predict the students in each group.

Exploring the Data

Some of the important characteristics of the data are given below:

Number of data points	395
Number of features	30
Number of graduates	265
Number of non-graduates	130
Graduation rate	67.09%

Training and Evaluation

1. Decision Tree Classifier

Decision tree classification is widely used and easy to interpret algorithm for supervised learning. A decision tree is represented as a binary tree whose nodes are the decisions on which data is split and leaves are the final classification. The data is split in a top down approach, with each node splitting the data on the basis of a metric such as Gini impurity and information gain. A tree is built in this way until a required depth is reached.

A decision tree is easy to interpret and if the number of variables are small it can be represented in a simple graphic. Decision tree algorithm works well with both numerical and categorical data. It also scales well and can be used to analyze large data sets with limited computational resources and time. This is the major reason due to which decision tree is chosen for this project. If a decision tree can be built with a reasonable accuracy, it is easy to implement on a large data set. One major disadvantage of decision tree is that it tends to overfit[1]. Decision tree algorithm can create complex trees from the training data that do not generalize well. Table 1 shows the F1 score and time taken for training and testing the student data using decision tree classifier. It can be seen that decision tree overfits the data as the F1 score is 1 in all the cases. This means that the training data fits perfectly with the decision tree classifier.

[1] http://stats.stackexchange.com/questions/1292/what-is-the-weak-side-of-decision-trees

Table 1: F1 score and time efficiency of Decision Tree classifier

Training Set Size	100	200	300
Training Time (s)	0.001	0.002	0.003
Prediction Time (s)	0.000	0.001	0.000
F1 Score (Training)	1.0	1.0	1.0
F1 Score (Testing)	0.693	0.677	0.703

2. Random Forest

Random forest is an ensemble learning method for classification. Random forest is a way to overcome the problem of overfitting in the decision tree classifier by taking the average of several different decision trees. For each decision tree, a subset of the data set is made with replacement. Then, *m* predictor variables are selected at random and a decision tree is built using the subset of data and variables. Prediction for a new data point is made by taking the majority vote of all the decision trees. One of the advantages of random forest is that they improve the accuracy of decision tree classifiers, at the cost of lower interpretability. They can be generalized to predict the labels of new data points. Like decision tree classifiers, random forests are time and space efficient and can be scaled easily. The main disadvantage of random forest classifier is that generating large number of tress can be slow[2]. Random forest has been chosen for this project to increase the accuracy of prediction with insignificant increase in costs. Table 2 shows the F1 score and time efficiency of random forest classifier in training and prediction of student data.

Table 2: F1 score and time efficiency of Random Forest classifier

Training Set Size	100	200	300
Training Time (s)	0.009	0.010	0.015
Prediction Time (s)	0.002	0.007	0.002
F1 Score (Training)	0.978	0.989	0.988
F1 Score (Testing)	0.768	0.711	0.743

3. Support Vector Machine

Support vector machine (SVM) is a supervised machine learning algorithm for classification. A SVM determines a hyperplane that can separate two groups of points. A simple SVM uses a linear separator but 'kernel' trick allows non-linear separator to be used. A SVM separator focuses only on the points from both the groups that are near to each other and are hard to separate. These points are called support vectors. SVM works by estimating the separator that maximizes the margin between support vectors.

Support vector machines solve a convex optimization problem and therefore, always converge. Their accuracy is also generally higher and can also be used with higher dimensional data[3]. The downside with

[2] http://www.nickgillian.com/wiki/pmwiki.php/GRT/RandomForests
[3] http://www.nickgillian.com/wiki/pmwiki.php/GRT/SVM

using SVMs is that they are slower and with large datasets, training SVMs can take a significantly larger time[4]. SVM require the data to be numeric whereas decision trees work with mixed data as well. SVM classifier is selected for this project as their prediction accuracy is generally higher and the data used in this project is numeric. Table 3 shows the accuracy and time efficiency of the SVM classifier.

Table 3: F1 score and time efficiency of SVM classifier

Training Set Size	100	200	300
Training Time (s)	0.001	0.005	0.011
Prediction Time (s)	0.001	0.002	0.004
F1 Score (Training)	0.866	0.878	0.874
F1 Score (Testing)	0.773	0.779	0.779

Choosing the Best Model

Table 1, Table 2 and Table 3 shows the F1 score and time efficiency of the three chosen models. From the tables, it is clear that the decision tree classifier is the most time efficient, even with larger training data size. However, it is not the most accurate. Random forest model has better accuracy than decision tree at the cost of higher computation time in training and prediction. Keeping in mind that the training and testing accuracy may depend on how the training and testing sets were partitioned, it is an indicator of how well the model will generalize. Support vector machine classifier comes out to be the most accurate model amongst the three. Also, in this scenario, accuracy should be given more weight than the time taken to make the prediction. School management using this model would want to get the prediction right for each student as lack of intervention could mean lower graduation rate in the future.

F1 score and time efficiency are also calculated by taking subsets of different sizes of training data (100, 200 and 300). In random forest and decision tree classifier, the testing F1 score is maximum when the size of training data is largest. In SVM, by increasing the training data size, there is insignificant increase in the prediction accuracy. In all the models, training time and prediction time increases with increase in the size of training data. Based on all these observations, it can be said that the SVM classifier is most appropriate for the task. SVM can give better accuracy scores with a smaller training size that the other two models. With a training size of 200, the testing F1 score of SVM is 0.779, whereas the same of decision tree is 0.677 and random forest's is 0.711. The training time and prediction time for SVM is higher than decision tree but given the higher accuracy at smaller training size, SVM is a more appropriate model.

If we plot the data as the set of points, Support vector machines work by estimating a plane that separate two sets of points. A simple two dimensional case[5] can be seen in the fig 1. The red dots belong to one group and blue dots belong to another, each dot being a data point plotted in two dimensions. Our goal is to predict the group of a new dot or data point given to us. SVM classifier comes up with a line that separates the two groups. SVM estimates the separator by calculating its distance with the points near to it that belong to each group. The group for a new data point is predicted by using

[4] http://stackoverflow.com/questions/18165213/how-much-time-does-take-train-svm-classifier

[5] https://www.reddit.com/r/MachineLearning/comments/15zrpp/please_explain_support_vector_machines_svm_like_i

this separator. If new data point is to the right of this separator, it belongs to the red group and to the blue group if it is to the left of the separator.

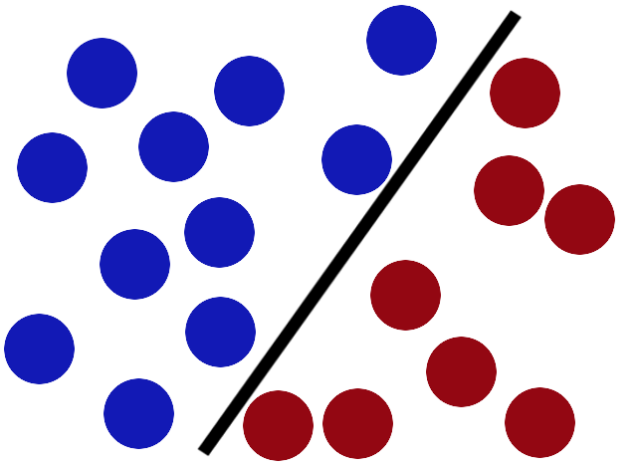

Figure 1: SVM with two dimensional data

This model can be extended to use a boundary that is curved to give it more flexibility to classify points that cannot be correctly classified by a straight line. This model can also be extended to higher dimensions effectively, such as the student data we are using. Training data is used to calculate such a boundary and testing data is used to measure the accuracy of the classifier.

Final Model

Grid search algorithm is used to find the optimum parameters for the SVM classifier. The parameters of SVM classifier that are optimized are: 'C' and 'gamma'. For larger values of C, the separating hyperplane will try to correctly classify all the training example by using a hyperplane with smaller margin. For smaller values, the classifier uses a large margin separating hyperplane even if it does not classify all the training examples correctly. Therefore, bias increases and variance decreases when C decreases. For smaller gamma values, the decision boundary is nearly linear. Increasing gamma also increases the flexibility of the separating hyperplane and thus leads to overfitting. The optimum parameters and resulting F1 score for the final model are given in the table below.

Optimum parameters	C	10.0
	Gamma	0.1
Training F1 score		0.826
Testing F1 score		0.792

References

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning*. New York: Springer.