Building a Student Intervention System: An Udacity Nanodegree ML Project

Omoju Miller

May 10, 2016

Introduction

Your goal is to identify students who might need early intervention - which type of supervised machine learning problem is this, classification or regression? Why?

- This task sounds like a problem that would be best suited for a classification algorithm. The inherent task is to develop a learners that can "predicting a category." If we look at the problem from another perspective, we can consider the student data available as a "labeled" dataset. We have features that we can use to determine who has succeeded in the class versus who has not. For that insight, we could use '`passed`' column as our class label.

Models

For the student intervention challenge, three supervised learning algorithms have been selected as appropriate learners for the task. The algorithms are as follows:

1. Decision tree classifier
2. Random forest classifier
3. Support vector machines

Decision Tree Classifier

- What is the theoretical $O(n)$ time & space complexity in terms of input size?
  The theoretical time & space complexity of decision trees classifiers as implemented in sci-kit learn package is $O(v_{features} n^2_{samples} log(n_{samples}))$.
- What are the general applications of this model?
  The decision tree algorithm is usually applied to classification and regression problems.
  What are its strengths and weaknesses?
  - Strengths of decision trees:

    * Very intuitive. You can look at the results and understand it.
    * Requires little data preparation.
    * The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. In average case the cost of training $O(pNlog^2N)$
    * Able to handle both numerical and categorical data.
    * Very robust. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.
  - Weaknesses of decision trees:
    * Decision-tree learners can create over-complex trees that do not generalize the data well. They are prone to over-fitting especially in the case of data with lots of features.
    * Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
    * The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts.
    * There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
    * Decision tree learners create biased trees if some classes dominate.

- Given what you know about the data so far, why did you choose this model to apply?
  The decision tree classifier was chosen because of its ease of use and its relatively cheap time complexity.
  Another *major* reason why the decision tree classifier was selected was its ease of interpretation of the classifier results. For this problem domain, it isn't just satisfactory to identify students that need intervention, what learning researchers ultimately want is to gain *insights* into the nature of learning, and the social factors that lead to certain outcomes for at-risk students. A decision tree learner, with its ability to graphically plot out the tree becomes a research tool in the hands of learning scientist. Consequently, this can help the school board of supervisors build better solutions for those students which well executed could potentially reduce the costs associated with remediating failed students.

Table 1: Result of training with a DecisionTreeClassifier

	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.002
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	1.000	1.000	1.000
F1 score for test set	0.737	0.775	0.690

Random Forest Classifier

- What is the theoretical $O(n)$ time & space complexity in terms of input size?

Table 2: DecisionTreeClassifier - Confusion Matrix

		Actual Class	
		Passed	Fail
Predicted Class	Passed	19.000	14.000
	Fail	22.000	40.000

The theoretical time & space complexity for building a complete unpruned decision tree is $O(v_{features} * n^2_{samples} log(n_{samples}))$. For a random forest with mtry as the number of features sampled at each node, the complexity would be $O(n_{estimators} * mtry * *n^2_{samples} log(n_{samples}))$

- What are the general applications of this model?
  Ensemble learners are used in supervised learning. The goal of the random forest classifier is to combine the results of multiple classifiers, it is an ensemble of decision trees. The prediction of the classifier is the averaged prediction of the individual classifiers. What are its strengths and weaknesses?
  - Strenghts of SVMs:
    * Fast to train
    * Flexible, can be used with large number of attributes, small or large datasets
  - Weaknesses of decision trees:
    * Loss of interpretability as compared to decision trees give.
    * Can be cost to train because of number of trees. However, the algorithm leands itself well to parallelization which significantly boosts the speed.
- Given what you know about the data so far, why did you choose this model to apply?
  This learners was chosen as a means of reducing the effects of overfitting of decision trees.

Table 3: Result of training with a Random Forest Classifier

	Training set size		
	100	200	300
Training time (secs)	0.361	0.370	0.387
Prediction time (secs)	0.011	0.011	0.012
F1 score for training set	1.000	1.000	1.000
F1 score for test set	0.805	0.814	0.783

Table 4: Random Forest Classifier - Confusion Matrix

		Actual Class	
		Passed	Fail
Predicted Class	Passed	11.000	22.000
	Fail	8.000	54.000

Support Vector Machine (SVMs)

- What is the theoretical $O(n)$ time & space complexity in terms of input size?
  The theoretical time & space complexity of SVMs scales between $O(n_{features} * n^2_{samples})$ and $O(n_{features} * n^3_{samples})$. Its a costly algorithm since the compute and storage requirements increase rapidly with the number of training vectors.
- What are the general applications of this model?
  SVMs are an algorithm used for classification, regression and outlier detection.
  What are its strengths and weaknesses?
  - Strenghts of SVMs:
    * Effective in high dimensional spaces.
    * Still effective in cases where number of dimensions is greater than the number of samples.
    * Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
    * Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.
  - Weaknesses of decision trees:
    * If the number of features is much greater than the number of samples, the method is likely to give poor performances.
    * SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
- Given what you know about the data so far, why did you choose this model to apply?
  The general approach is to try a few different algorithms for each problem. Sci-kit learn provides a guide to choosing estimators. Based on the guide, and satisfying the following criteria: 100K >samples >50, Predicting a category & Labeled data, a linear SVC was selected.

Table 5: Result of training with a SVMs

	Training set size		
	100	200	300
Training time (secs)	0.005	0.018	0.025
Prediction time (secs)	0.000	0.001	0.001
F1 score for training set	0.916	0.826	0.851
F1 score for test set	0.713	0.738	0.738

Table 6: SVMs - Confusion Matrix

		Actual Class	
		Passed	Fail
Predicted Class	Passed	13.000	20.000
	Fail	14.000	48.000

Best Model

Based on the experiments you performed earlier, in 1-2 paragraphs explain to the board of supervisors what single model you chose as the best model. Which model is generally the most appropriate based on the available data, limited resources, cost, and performance?

In 1-2 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work (for example if you chose a Decision Tree or Support Vector Machine, how does it make a prediction).

Fine-tune the model. Use Gridsearch with at least one important parameter tuned and with at least 3 settings. Use the entire training set for this. What is the model's final F1 score?

Conclusion

This paper has laid out some of the challenge of