4. Training and Evaluating Models

When we look at the data, we can see that the problem is classification problem. Therefore, I have chosen Decision Tree, SVM and Naïve Bayes supervised learning algorithms. Below I will explain strengthens and weaknesses of these models.

Decision Tree is the first model I experimented. Strengthens of decision tree:

- Computational complexity is low. Since it is a binary tree. Complexity of decision tree is O(n*log(n))
- Decision tree is easy to use and easy to explain to somebody.

Weaknesses of decision tree:

- Decision Tree can easily overfit. We can see it in below comparison tables. Decision Tree learns all data in training and it got best F1 score for training data, but F1 score is not good for testing data. In fact, when we increase training set sizes F1 score for test data is decreasing.
- Decision Tree does not work if there are a lot of unrelated variables.

Training Set 100 (Decision Tree)			
training time	prediction time	F1 score on training set	F1 score on test set
0.001	0.000	1.0	0.708661417323

Training Set 200 (Decision Tree)			
training time	prediction time	F1 score on training set	F1 score on training set
0.002	0.000	1.0	0.693548387097

Training Set 300 (Decision Tree)			
training time	prediction time	F1 score on training set	F1 score on test set
0.002	0.000	1.0	0.688524590164

Second algorithm I chose is Support Vector Machine (SVM). Advantages of SVM:

- SVM is robust to overfitting. As we see in below comparison tables. When we increase training data size, F1 score for training data does not change much.
- SVM can model non-linear relations.
- Effective in higher dimensional spaces

Weaknesses of SVM:

- Computational complexity is higher than Decision Tree.

Training Set 100 (SVM)			
training time	prediction time	F1 score on training set	F1 score on test set
0.001	0.001	0.86301369863	0.794520547945

Training Set 200 (SVM)			
training time	prediction time	F1 score on training set	F1 score on test set
0.003	0.003	0.863636363636	0.820512820513

Training Set 300 (SVM)			
training time	prediction time	F1 score on training set	F1 score on test set
0.009	0.006	0.861538461538	0.838709677419

Third algorithm I chose is Naïve Bayes. Strengths of Naïve Bayes:

- The model is simpler.
- Computational complexity is less than SVM.
- Fast to train and fast to classify

Weaknesses of Naïve Bayes:

- Assumes that all features are independent

Training Set 100 (Naïve Bayes)			
training time	prediction time	F1 score on training set	F1 score on test set
0.001	0.000	0.815384615385	0.77519379845

Training Set 200 (Naïve Bayes)			
training time	prediction time	F1 score on training set	F1 score on test set
0.001	0.000	0.786764705882	0.772727272727

Training Set 300 (Naïve Bayes)			
training time	prediction time	F1 score on training set	F1 score on test set
0.004	0.002	0.797136038186	0.782608695652

5. Choosing the Best Model

As we see in previous section, each model has strengthens and weaknesses. If we are searching the best model in terms of computationally cost, then Decision Tree is the best model. In terms of accuracy, SVM is the best model. Since data is small and complexity of SVM is not issue here. Therefore, SVM is the best single model for this problem.

In laymen terms we need to find two groups of students, one which can pass, another one which can not pass the class. When we look at the data we can see that there are certain characteristics of these groups. If we learn these characteristics, then we can predict in the future student will fail or pass. When we learn these characteristics, we can even find a separator to separate these groups. This separator can we be linear or nonlinear based on data. And we find a separator we try to have a maximum margin between these two groups. Support Vector Machines (SVM) does exactly above what we have explained.

After fine-tune the model, I got F1_score=0.88.