Student Intervention judgement

Kosuke Fukui

1. Classification or Regression

In this problem, we'd like to know whether student pass the final exam or not. Therefore, I use classification method. Classification can be applied to the problem of which the answer belongs to particular class like "agree or disagree", "A team or B team or C team" and "pass the exam or not".

2. Evaluating Model Performance

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

3. Preparing the data

I used *train_test_split* from *sklearn* for random data assignment for training and test dataset. I adopted *label_binarize* from *sklearn* to binarize the target column from "yes and no" to "0 and 1".

4. Training and Evaluating Models

(1) Decision Tree
Select Reason: The structure is simple and easy to implement.
Application: Buyer behavior analysis.
Strengths: The result tree can be visualized and easy to interpret it. The condition can be explained by boolean logic, therefore the model is easy to understand.
Weaknesses: The result can be changed easily by a little change of the data. It is difficult to find the optimal parameters because this is one of the NP-complete problems.

Decision Tree	Training data size		
	100	200	300
Training time	0.001	0.001	0.003
Prediction time	0.000	0.000	0.001
F1 for training data	1.000	1.000	1.000
F1 for test data	0.729	0.708	0.695

(2) Random Forest

Select Reason: High prediction performance.The structure is simple and calculation speed is high.
Application: This method used for image recognition on medical fields.
Strengths: High complexity model can be used in low variance because random forest uses ensemble methods.
Weaknesses: The model tends to be complicated because this model uses various decision trees and the number of the parameters can be large. In small data, this model doesn't work well because this model picks training data and variables randomly.

Random Forest	Training data size		
	100	200	300
Training time	0.031	0.025	0.025
Prediction time	0.001	0.002	0.001
F1 for training data	0.992	1.000	0.993
F1 for test data	0.782	0.797	0.761

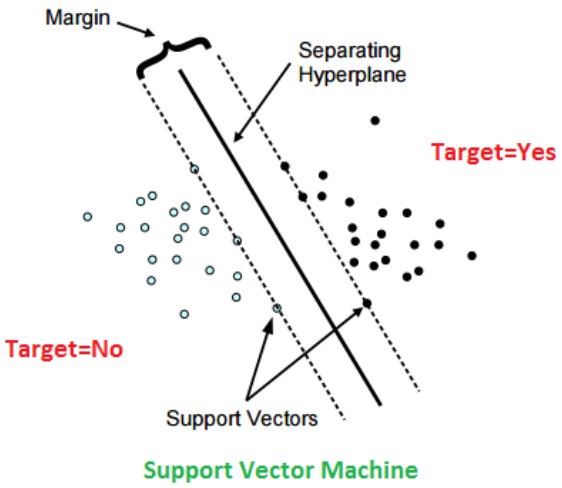
(3) Support Vector Machine
Select Reason: It is known that this method shows high predictability on various problems.
Application: This method often used for text processing like detecting spam mail.
Strengths: High predictability because of high generalization ability. By Kernel trick, this method works when the dimensions is larger than the number of samples.
Weaknesses: Training time is long on big data size.

SVM	Training data size		
	100	200	300
Training time	0.024	0.004	0.027
Prediction time	0.002	0.003	0.006
F1 for training data	0.859	0.869	0.869
F1 for test data	0.784	0.776	0.759

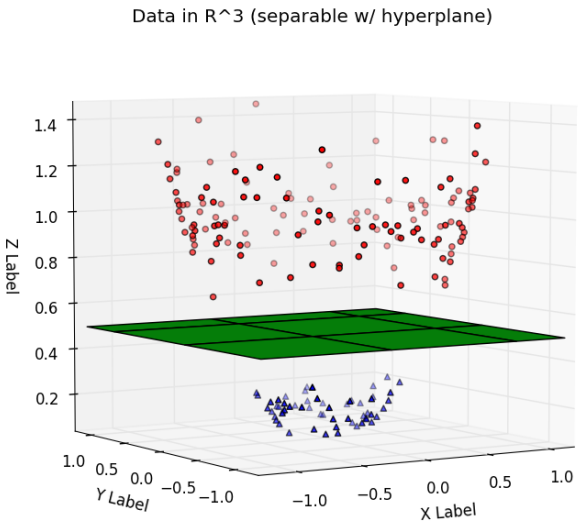
5. Choosing the Best Model

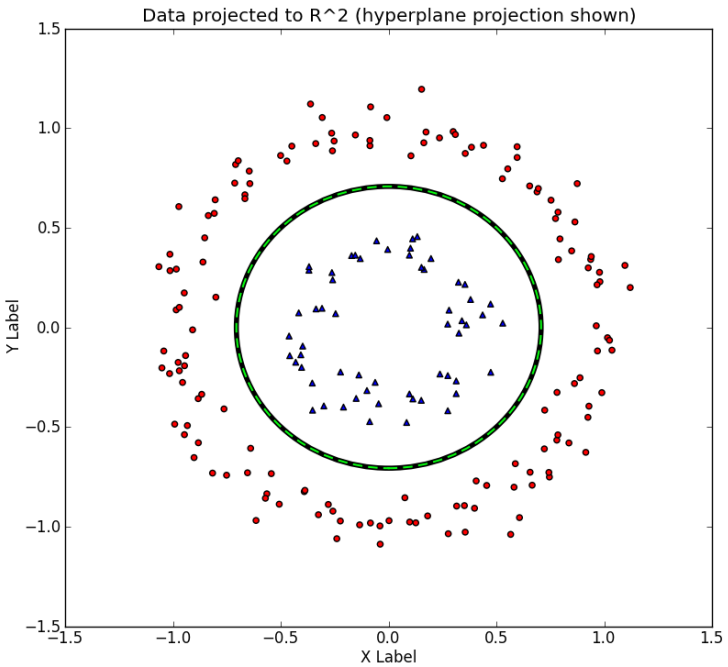
I chose Support Vector Machine as the best model. Decision Tree is simple but that shows low F1 score and it doesn't have plenty tuning potential. Random Forest shows good F1 score but it tends to take more training time than SVM.

SVM is a process to find a line that separates between the different groups, for instance "Yes" or "No". SVM chooses the best line where the distance between the line and the nearest data points of different groups are the largest. The distance called margin and the nearest points called support vectors. SVM predicts a new data is included to which group based on the separating line.



Moreover, the powerful feature of SVM is that this method can separates the data not only by simple linear line but also by non-linearly line. When a problem is difficult to separate data by linear line, the problem can be easy in higher dimensions as below. The right figure is shows that a problem is difficult to separate data by single linear line, but in 3 dimensions the data can be divided easily as the left figure shows. This is called karnel trick and strengthen the application potentiality of SVM.





The final F1 score applied GridSearchCV is as followings:

Best parameter choice: {'kernel': 'rbf', 'C': 100, 'gamma': 0.0001}

Best F1 score: 0.824511363636