Student Intervention Report

Xiangwei Wang wangxiang.cpp@gmail.com
April 1st 2016

1. Classification VS Regression

Question 1: Your goal is to identify students who might need early intervention - which type of supervised machine learning problem is this, classification or regression? Why?
Answer 1:We should use classification, because the target is to correctly identify to know whether or not a student need early intervention. It is classification problem, and regression problem is with continue labels such as suit data with a line model and so on.

2. Exploring the Data

- Total number of students: 395
- Number of students who passed: 265
- Number of students who failed: 130
- Number of features: 30
- Graduation rate of the class: 67%

3. Prepare the Data

1. feature column: 'school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob', 'reason', 'guardian', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences'
2. target column: passed

4. Training and Evaluating Models

I choose Gauss Native Bayes, Decision Tree and Adaboost

4.1 the general application and its strengths, weaknesses?

1.Naive Bayes

General Application

Classification, and have a good performance even though there are big feature spaces.

Strengths

1. easy to implement
2. deal with big feature space
3. efficient

weaknesses

1. Naive Bayes model assumes that each of the features used are conditionally independent of one another given some class. It is a theoretical weakness of Native Bayes, but sometimes, the performance is good though features are not conditionally independent of one another given some class in practice.

2.Decision Tree

General Application

Nonlinear Classification ot regression, some circumstance when we need interpret how the classification work and which the feature classification use. **Strengths**

1. easy to use
2. graphically, interpret

Weaknesses

1. easy to overfit without good parameter

3.Boosting
General Application

Robust classification with little possibility to be overfitting.

Strengths

1. Computationally efficient.
2. No difficult parameters to set.
3. Versatile a wide range of base learners can be used with AdaBoost.

Weaknesses

1. Algorithm seems susceptible to uniform noise.
2. Weak learner should not be too complex to avoid overfitting.
3. There needs to be enough data so that the weak learning requirement is satisfied the base learner should perform consistently better than random guessing, with generalization error < 0.5 for binary classification problems.

4.2 Why choose this model?

There are 30 attributes and only 395 datas intotal. And it is very high dimention classification, I abandon SVM,NN and Instance based learning because of the curse of dimention. And as a result, I choose native bayes, decision tree and adaboost which have a better performance with high dimention than SVM,NN,and Instance based learning.

4.3 Table

Gauss Native Bayes

Training Set size	Training Time	Predicting Time(Testing)	Traing score	Testing Score
100	0.001s	0.000s	0.855	0.748
200	0.001s	0.000s	0.832	0.713
300	0.002s	0.000s	0.809	0.75

Decision Tree without default Parameter(If None, then nodes are expanded until all leaves are pure)

Training Set size	Training Time	Predicting Time(Testing)	Traing score	Testing Score
100	0.001s	0.000s	1.0	0.683
200	0.001s	0.000s	1.0	0.737
300	0.002s	0.000s	1	0.706

Adaboost classifier with default Parameter(base estimator = DecisionTreeClassifier, n_estimators=50, learning_rate=1)

Training Set size	Training Time	Predicting Time(Testing)	Traing score	Testing Score
100	0.096s	0.004s	0.954	0.72
200	0.084s	0.004s	0.883	0.806
300	0.096s	0.004s	0.868	0.779

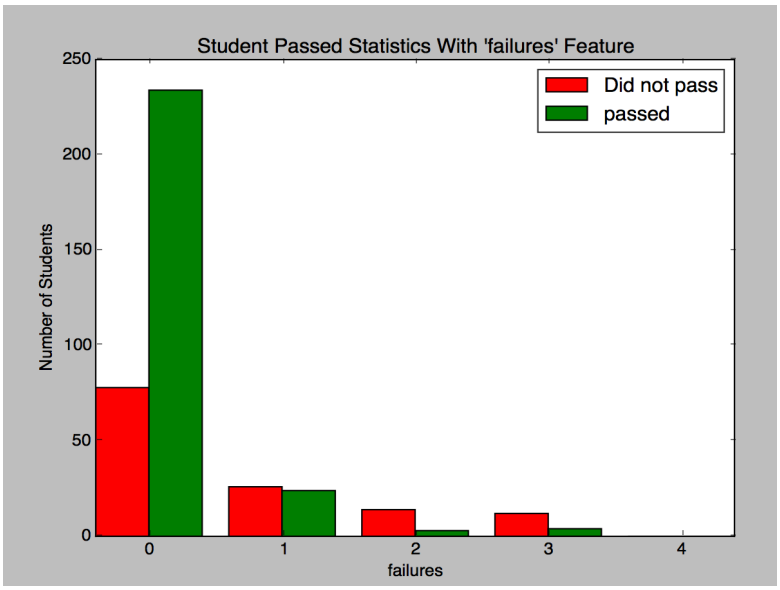
5. Choosing the Best Model

5.1 The best model

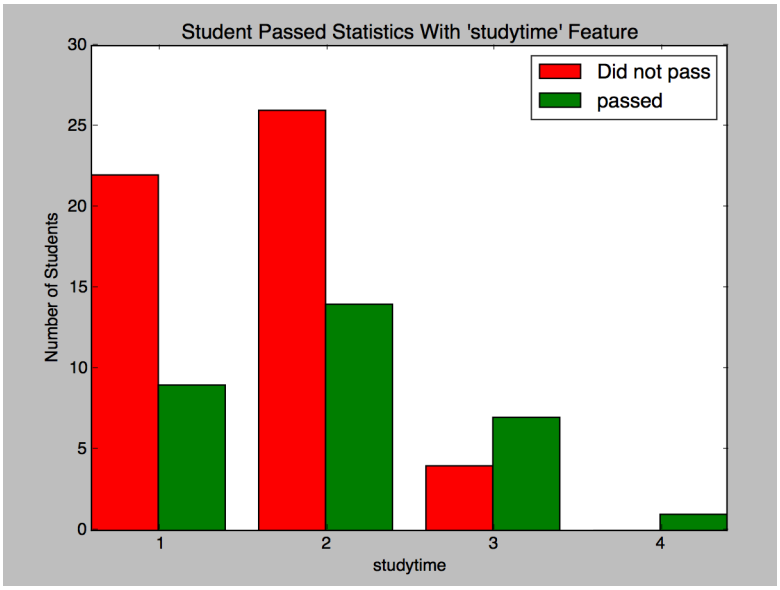
I choose Adaboost as the best model.
Obviously Adaboost has the best testing F1 score as we all know that what matters is not training score but testing score, so we do not care decision tree's 100% score in training which is result of overfiting.
Decision tree and Gauss Native Bayes use less time than Ababoost both in training and testing. But because the size of data is limited, it is okey for Adaboot using more time.
In conclusion, I choose Adaboost to be the best model for accuracy.

5.2 How does the best model work?

We use adaboost learning model to learn and judge whether a student need early intervention. Firstly we find some basic rules between the type of student and whether the student may not pass the final exam. The rules maybe "a student who has failed one or more time in the past exam may not passed exam this time", or "student who want higher education can pass the final exam" and so on. We get the first rule which can do a better job than guess easily, a rule just have over 50% possibility to be right is okey and our first rule is "a student who has failed one or more time in the past exam may not passed exam this time" and the following chart shows that the rule really do better than chance;



Then we concentrate more on the wrong result by the first rules: there are some students pass the final exam though they did not passed in previous exam and some other student do pass previous exam but not pass this time, we should pay more attention to these kinds of students when find the second rule,and we find that some student who did not pass before use more time to study and they pass the exam, so we get the second rule "A student with more study time(more than 5 hours) can pass the exam";



and similarly, we obtain enough rules, then the final rules the combine all the rules we got if more rules indicate a student can pass the final exam then we think he can, of course not all rules has the similar importance. The rule with higher score when found plays a more important role! What is amazing is that all the process is done by our model with human interrupt!
This is how our model works!

5.3 final F1 score

The final F1 score is 0.789115646259(Adaboost, n_estimator = 10, learning_rate=0.5)

Reference

[1] Scikit-learn http://scikit-learn.org/stable/index.html
[2] Inroduction to Boosting
https://storage.googleapis.com/supplemental_media/udacityu/5435300514/Intro%20to%20Boosting.pdf
[3]Difference between naive Bayes & multinomial naive Bayes
http://stats.stackexchange.com/questions/33185/difference-between-naive-bayes-multinomial-naive-bayes