

Machine Learning Engineering Nanodegree

Project 2: Building a Student Intervention System

version 0.4

Jouni Huopana
12/28/2015

CONTENTS

0. Introduction	2
1. Classification vs Regression	2
2. Exploring the Data	4
3. Preparing the Data.....	5
4. Training and Evaluating Models	6
Decision tree model	6
Random forest model.....	6
Naïve Bayes model.....	6
5. Choosing the Best Model	8

0. INTRODUCTION

This document contains information relating to the Student Intervention System dataset, which is used for the development of a machine learning model to help estimate if students pass their final high school exam. Basic statistics of the data are derived and used then to create models to make predictions on that data and then finally a cross validated prediction model is created. The model is then used to predict for a given input and the model performance is discussed. The Python code can be found in accompanied iPython notebook – file (student_intervention.ipynb). This document has been created as a project work for the Udacity Machine Learning Engineer Nanodegree.

1. CLASSIFICATION VS REGRESSION

The goal is to create an intervention system for students to indicate if the student in questions is likely to pass the high school final exam. The given data includes information about the background of the students and their behavior. More detailed descriptions can be seen in table 1. As the outcome of the model is determined either yes or no (passed/failed), the sensible thing to do is to use binary classification as the basis of this model. If the data would include more time dependent data such as monthly data or target would be an actual grade, it could be argued that a regression model could also work.

Table 1 Dataset attributes

Attributes for student data.csv:	
school	student's school (binary: "GP" or "MS")
sex	student's sex (binary: "F" female or "M" male)
age	student's age (numeric: from 15 to 22)
address	student's home address type (binary: "U" urban or "R" rural)
famsize	family size (binary: "LE3" less or equal to 3 or "GT3")
Pstatus	parent's cohabitation status (binary: "T" living together or "A" apart)
Medu	mother's education (numeric: 0 none, 1 primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
Fedu	father's education (numeric: 0 none, 1 primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education)
Mjob	mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
Fjob	father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
reason	reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
guardian	student's guardian (nominal: "mother", "father" or "other")
	home to school travel time (numeric: 1 <15 min., 2 15 to 30 min., 3 30 min. to 1 hour, or 4 >1 hour)

traveltime	
	weekly study time (numeric: 1 <2 hours, 2 2 to 5 hours, 3 5 to 10 hours, or 4 >10 hours)
studytime	
failures	number of past class failures (numeric: n if $1 \leq n < 3$, else 4)
	extra educational support (binary: yes or no)
schoolsup	
famsup	family educational support (binary: yes or no)
paid	extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
activities	extra curricular activities (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
internet	Internet access at home (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
famrel	quality of family relationships (numeric: from 1 very bad to 5 excellent)
freetime	free time after school (numeric: from 1 very low to 5 very high)
goout	going out with friends (numeric: from 1 very low to 5 very high)
Dalc	workday alcohol consumption (numeric: from 1 very low to 5 very high)
Walc	weekend alcohol consumption (numeric: from 1 very low to 5 very high)
health	current health status (numeric: from 1 very bad to 5 very good)
absences	number of school absences (numeric: from 0 to 93)
passed	did the student pass the final exam (binary: yes or no)

Also it should be noted that if some of these values are taken during the education process, it is likely that if the model is taken into use, then the model becomes invalid in the future. Only if the input data can be gathered before the actual education process begins, we can develop a model which predicts the outcome.

2. EXPLORING THE DATA

The dataset has the following overall features, which are gathered from the dataset:

- 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%

The number of passed is somewhat unbalanced when compared to the failed. More even dataset could provide better model and also it would help if the dataset would be larger. Sample of the first few dataset values are shown in table 2 and example box plots in figure 1.

Table 2 Dataset sample

School	GP	GP	GP	GP	GP	GP	...
sex	F	F	F	F	F	M	...
age	18	17	15	15	16	16	...
address	U	U	U	U	U	U	...
famsize	GT3	GT3	LE3	GT3	GT3	LE3	...
Pstatus	A	T	T	T	T	T	...
Medu	4	1	1	4	3	4	...
Fedu	4	1	1	2	3	3	...
Mjob	at_home	at_home	at_home	health	other	services	...
Fjob	teacher	other	other	services	other	other	...
reason	course	course	other	home	home	reputation	...
guardian	mother	father	mother	mother	father	mother	...
traveltime	2	1	1	1	1	1	...
studytime	2	2	2	3	2	2	...
failures	0	0	3	0	0	0	...
schoolsup	yes	no	yes	no	no	no	...
famsup	no	yes	no	yes	yes	yes	...
paid	no	no	yes	yes	yes	yes	...
activities	no	no	no	yes	no	yes	...
nursery	yes	no	yes	yes	yes	yes	...
higher	yes	yes	yes	yes	yes	yes	...
internet	no	yes	yes	yes	no	yes	...
romantic	no	no	no	yes	no	no	...

famrel	4	5	4	3	4	5	...
freetime	3	3	3	2	3	4	...
goout	4	3	2	2	2	2	...
Dalc	1	1	2	1	1	1	...
Walc	1	1	3	1	2	2	...
health	3	3	3	5	5	5	...
absences	6	4	10	2	4	10	...
passed	no	no	yes	yes	yes	yes	...

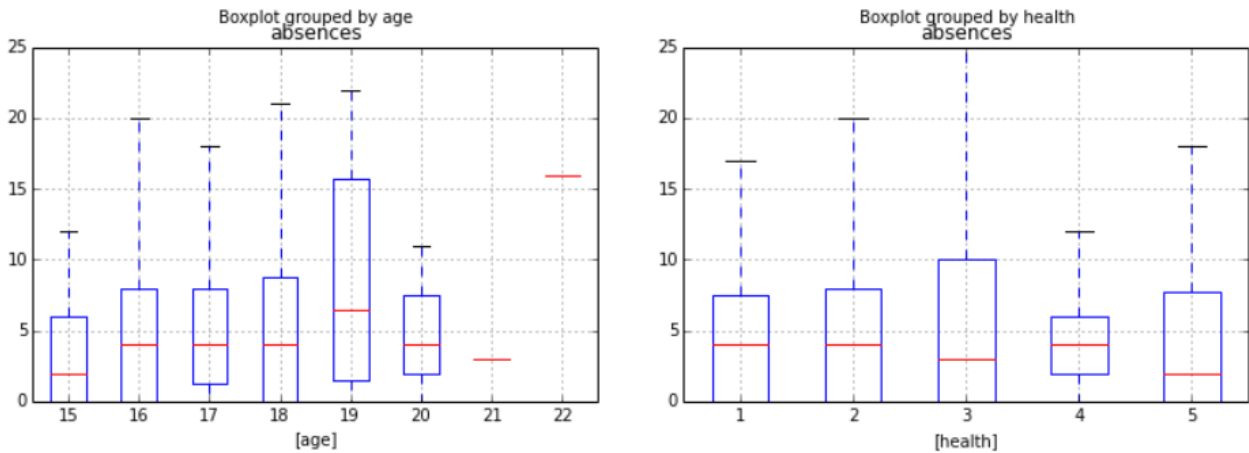


Figure 1 Example box plots of the data

3. PREPARING THE DATA

The given data set is transformed into numerical data. This means that feature such which have 'yes' or 'no' are transformed to 1 and 0. If there are more than two variables in feature an extra column is created to include this data. Also the target value of the "passed" is extracted and transformed in the similar manner. After all the columns have been created, the data is split into training and test sets. Training set contains 300 rows and testing set 95 rows. The training set is also split into three subsets where number of rows are 100, 200, 300. The last subset is identical to whole training data. These datasets are used without changes throughout this project.

4. TRAINING AND EVALUATING MODELS

Three models were chosen to be evaluated for this case

- Decision trees
- Random Forests
- Naive Bayes

All of the selected models can do both classification and regression, but as it was discussed earlier, in this case classification is used to predict the outcome of the model

DECISION TREE MODEL

Decision tree learning uses a decision tree as a predictive model, which is widely used in statistics and machine learning. The model sets for each feature a limit which determines in which direction the model takes the following step. For example is feature "age" greater than 15. Each step takes the model closer to the final target (class). This model was chosen as the data has some features which resemble data that could be seen to have clear limits as age and features which are limited to discrete sets as 'yes' or 'no'. Decision trees are interpretable, easy to implement, scalable and fast to train, but can suffer from overfitting. The training information for this model can be found in table 4.

RANDOM FOREST MODEL

Decision trees use the data to determine the mathematical "best" place to set the limits for each decision. Very deep decision trees can overfit quite easily. To fight overfitting random forest models were introduced. Random forest averages multiple trees to reduce variance in the model. This has been shown to increase performance greatly. Random forest model has largely the same positive features than decision trees, but random forests are harder to interpret. This model was chosen as the data can sometimes have features which are difficult to interpolate with precise mathematical ways or limit to certain models. The training information for this model can be found in table 5.

NAÏVE BAYES MODEL

Naive Bayes classifiers are a group of probabilistic classifiers. Based on Bayes' theorem it is assumed that the features in the data are not dependent from each other. As a mathematical method this is a simple and fast theorem to apply to any dataset. It however can require larger datasets than some other methods and assumes that the features are independent. This model was chosen as it is known to be fast to use, even with real time data. In this case the Gaussian Naïve Bayes model was selected, which also assumes that the data is Gaussian. The training information for this model can be found in table 3.

Using the given dataset, the three selected models are tested against three different training sets. The training datasets have 100, 200 and 300 samples. The test set has 95 samples. Results from each test are shown in tables 3 -5.

Table 3 Decision Tree Testing

Training set size	100	200	300
Training time (secs)	0,004592	0,010768	0,017019
Prediction time (secs)	0,00028	0,000385	0,000429
F1 score for training set	1	1	1
Prediction time (secs)	0,000307	0,000271	0,000184
F1 score for test set	0,731343	0,782609	0,808511

Table 4 Random Forest Testing

Training set size	100	200	300
Training time (secs)	0,025808	0,030942	0,037617
Prediction time (secs)	0,002099	0,002412	0,002819
F1 score for training set	0,984615	0,984252	0,997455
Prediction time (secs)	0,001925	0,002026	0,002169
F1 score for test set	0,805755	0,814286	0,816901

Table 5 Naive Bayes Testing

Training set size	100	200	300
Training time (secs)	0,001863	0,002719	0,002577
Prediction time (secs)	0,000956	0,001431	0,001505
F1 score for training set	0,787879	0,765625	0,766585
Prediction time (secs)	0,00085	0,000841	0,000885
F1 score for test set	0,8	0,826087	0,828571

5. CHOOSING THE BEST MODEL

When testing the three different models, the training and test sets are identical in each corresponding case. From the tables 3-5 it can be seen that all of the models predict the test set fairly well, reaching the 0.8 with the largest train set. Also, it is clear that the Naïve Bayes model is the fastest to train due to its simplicity and that the Decision Tree model predicts the training data perfectly. The Random Forest model produces the high F1 scores when the whole dataset was used and also shows small overall relative increase in training time when compared to other models. The prediction times are short enough to be used in daily activities. All models can be trained under a second. Depending on the final size of the sample and training data, it is recommended to use the Naïve Bayes model for the student intervention model, as it gives good prediction and is the fastest in training and prediction, making it the cheapest to implement.

Naïve Bayes model uses probabilities to estimate if the features, such as in this case, 'Absence' or 'Health' indicates that the student will pass or fail. It is *naïve* as it each feature is considered independent. All the train data is used to create a combined probability to give a prediction if the student is in a need of intervention.

To find the best model, grid searching with 10 fold cross validation is used to find the optimal parameters for the model and to fight overfitting. In grid search different combinations of parameters can be used to create a model and the performance of that model is compared to other models. In the case of Naïve Bayes however, there is no parameters to fine-tune, but the cross validation is still done. The cross validation uses part of the train data to train the model and uses the rest of the data to estimate the performance. The model that gives the highest F1 score is finally chosen, basically picking the best of the bunch. The final model takes the given data and uses the best parameters to predict if the student passes or fails the final exams.

The final models F1 scores are:

- F1 score for training set: 0.77
- F1 score for test set: 0.83

This interestingly shows that the final did not improve in the terms of F1 score in the cross validation. This confirms that the chosen model works well and fast and can provide fast results when estimating student's changes to pass the final exam. Table 6 shows the predicted values against the true values. Totally 62 fails are predicted correctly and 214 passes are predicted correctly.

Table 6 Confusion matrix for the whole dataset

	Prediction Fail	Prediction Passed
True value, Fail	62	68
True value, Passed	51	214