Supervised Learning:
Building a Student Intervention System

Robert M Salom

June 17, 2017

1 Obective

The goal of this project is to identify students who might need early intervention. We want to predict whether a given student will pass or fail based on information about his life and habits. Therefore we approach this task as a classification problem with two classes, pass and fail.

2 Dataset

In what follows we will be working with part of the Student Performance Data Set from the UCI machine learning repository. It is composed of 395 data points with 30 attributes each. The 31'st attribute indicates whether the student passed or failed. Here is a brief description of each feature:

2.1 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" - greater than 3)
- 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")
- traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)

- studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
- failures - number of past class failures (numeric: n if 1<=n<3, else 4)
- schoolsup - extra educational support (binary: yes or no)
- 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)

2.2 Load the data

2.2.1 Imports

The following python libraries are used in this analysis.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tabulate import tabulate
```

2.2.2 Pre-process

Lets first observe what the graduation rate is for the class;

```
student_data = pd.read_csv("student-data.csv")

n_students = student_data.shape[0]
n_features = student_data.shape[1] - 1
n_passed = sum([1 for y in student_data['passed'] if y == 'yes'])
n_failed = sum([1 for n in student_data['passed'] if n == 'no'])
grad_rate = 100.*n_passed/(n_passed + n_failed)


tabulate( [["Total number of students: ",n_students],
```

```
["Number of students who passed: ",n_passed],
["Number of students who failed: ",n_failed],
["Number of features: ",n_features],
["Graduation rate of the class:",  "{:.2f}%".format(grad_rate)]], tablefmt="grid")
```

This gives us the following figures;

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:	67.09%