Build a Student Intervention System

1. Classification vs Regression

Your goal is to identify students who might need early intervention - which type of supervised machine learning problem is this, classification or regression? Why?

Classification, since the targets are divided into two classes: **passed** and **failed.**

2. Exploring the Data

Can you find out the following facts about the dataset?

- *Total number of students*
- *Number of students who passed*
- *Number of students who failed*
- *Graduation rate of the class (%)*
- *Number of features (excluding the label/target column)*

Use the code block provided in the template to compute these values.

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%

3. Preparing the Data

Execute the following steps to prepare the data for modeling, training and testing:

- *Identify feature and target columns*
- *Preprocess feature columns*
- *Split data into training and test sets*

Starter code snippets for these steps have been provided in the template.

Feature columns: 'school_GP', 'school_MS', 'sex_F', 'sex_M', 'age', 'address_R', 'address_U', 'famsize_GT3', 'famsize_LE3', 'Pstatus_A', 'Pstatus_T', 'Medu', 'Fedu', 'Mjob_at_home', 'Mjob_health', 'Mjob_other', 'Mjob_services', 'Mjob_teacher', 'Fjob_at_home', 'Fjob_health', 'Fjob_other', 'Fjob_services', 'Fjob_teacher', 'reason_course', 'reason_home', 'reason_other', 'reason_reputation', 'guardian_father', 'guardian_mother', 'guardian_other', 'traveltime', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences'

Target column: 'passed'

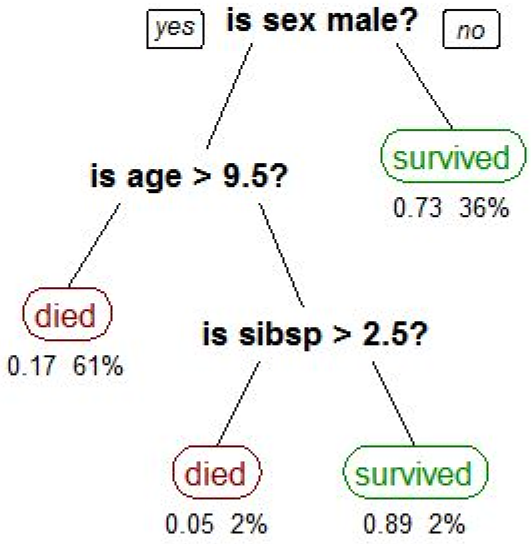
4. Training and Evaluating Models

Note: Since we want to get a measure of performance, I use the prediction time on training set instead of test set, for the reason that test set sizes are small, which makes their differences insignificant.

Model 1: Decision Tree Classifier

What are the general applications of this model? What are its strengths and weaknesses?

Commonly used in data mining, decision tree model maps possible classes into its leaves and features in its nodes.



Amongst its list of strengths are its simplicity and requires little data preparation. Its main weakness is its inability to represent certain, albeit relatively simple, concepts, such as XOR, parity, or multiplexer problems.

Given what you know about the data so far, why did you choose this model to apply?

Decision tree is usually my first choice in spot testing machine learning algorithms, since it is simple to understand, and if required it allows me to get list of feature importances. I use feature importances to see if there are features that are so negligible we don't need to use them, as sometimes their inclusion causes the final algorithm to generalize worse than without, in some models. In this project feature importances is not used as it is not part of the rubric specifications anyway.

In addition, I see that many of the features are binary in their values, which are easily representable using decision trees. This can be somewhat shown in perfect F1 scores when classifying training sets.

Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.002
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	1.0	1.0	1.0
F1 score for test set	0.654	0.725	0.661

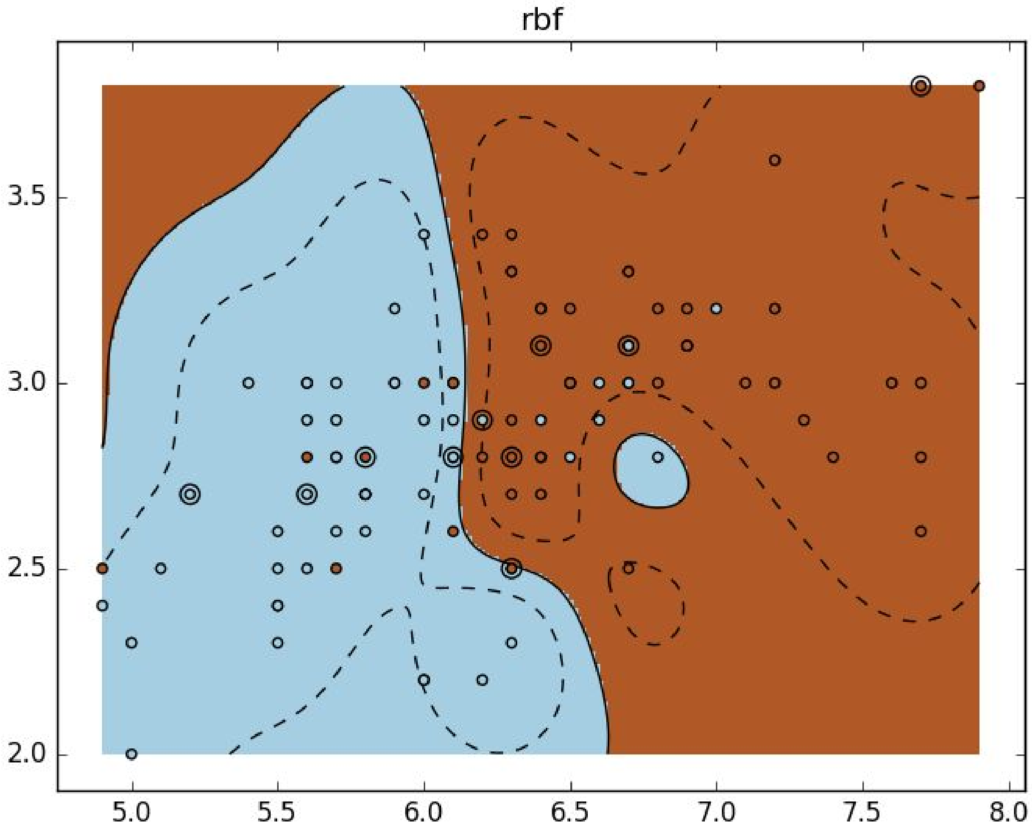
Model 2: Support Vector Classification

What are the general applications of this model? What are its strengths and weaknesses?

A variant of Support Vector Machine for classification problem. This model uses kernel trick to transform the given data to "extract" boundaries that exist among them.

This model is used across many different problems such as text and image classifications and really every other machine learning task. A simple modification to this model allows it to be used in unsupervised learning problem, that is to decide the right hypothesis to split the data points evenly across multiple classes.

It's main strength lies in its ability to understand complex concepts by drawing non linear boundaries. This is easier to be explained by an image. This image shows an example of this algorithm for deciding between two classes, taken from one of scikit learn's examples for SVC. Notice how it is able to draw complex boundaries:



Whereas in Decision Tree classifier the boundaries would look like boxes instead of polygons.

Given what you know about the data so far, why did you choose this model to apply?

There was no particular reason for choosing this model, I just wanted to see how it performed compared to decision tree. Due to the more complexity this model allows to represent turned out, as expected, we got better F1 score when predicting test set.

One thing to note here is that I needed to use feature scaling for data preprocessing. It is important to use since the model uses Euclidean distance to decide which classes the data points are in, which means features with larger differences like absences (value ranges between 0 to 75) will be treated differently than other features with small distances, e.g. all the binary features.

Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.004	0.010
Prediction time (secs)	0.001	0.003	0.000
F1 score for training set	0.802	0.827	0.812
F1 score for test set	0.803	0.777	0.774

Model 3: Multinomial Naive Bayes Classifier

What are the general applications of this model? What are its strengths and weaknesses?

A classifier that uses naive independence assumptions between the features to make its predictions. Multinomial Naive Bayes is generally used in text analysis / natural language processing.

Given what you know about the data so far, why did you choose this model to apply?

30 is a relatively large number of features for only 395 data points, even with only 2 classes (passed and not passed). As mentioned above, this means for a machine learning model to perform well, it will need much more data (2^30 or 1,073,741,824 to be exact!). Therefore it's only proper to choose a model that best handled that.

I chose multinomial over gaussian naive bayes simply because the latter performed better when spot tested without any parameter. I understand that a more proper way to do this is by plotting the data and see if they follow a gaussian distribution, but, you see, it's not an extra work I'm willing to do for this project.

Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.

	Training set size		
	100	200	300
Training time (secs)	0.001	0.001	0.002
Prediction time (secs)	0.000	0.000	0.000
F1 score for training set	0.805	0.848	0.810
F1 score for test set	0.806	0.800	0.806

5. Choosing the Best Model

Based on the experiments you performed earlier, in 2-3 paragraphs explain to the board of supervisors what single model you choose as the best model. Which model has the best test F1 score and time efficiency? Which model is generally the most appropriate based on the available data, limited resources, cost, and performance? Please directly compare and contrast the numerical values recorded to make your case.

From the higher F1 score alone, I choose Multinomial Bayes Classifier as the best model. As a bonus, it seems it has lower training and prediction time compared to the other two models, thus it will scale better for larger datasets.

To compare, the time required for training this classifier on 300 data points can take a fifth the time needed by one of the two classifiers we have tested. Multinomial Naive Bayes took around 0.002 seconds, similar with Decision Tree Classifier, while it took 0.010 seconds for SVC.

In 1-3 paragraphs explain to the board of supervisors in layman's terms how the final model chosen is supposed to work (for example if you chose a decision tree or support vector machine, how does it learn to make a prediction).

Naive Bayes is a classifier that uses naive independence assumptions between the features to make its predictions. That means, for example if students' "passed" depends on the features "availability of the internet" and their "number of absence", it is assumed that these two features do not affect each other.

It is trained by getting prior probabilities of all classes. With the same example, if we found that there are 80% of students with internet access passed, and and 30% of students with number of absences 10 passed, then the joint probability of these multiple evidences would be: **24% of**

students with internet access and number of absences equal to 10 passes this class. Basically the training gets all of these knowledge from previous experience / training set.

To make predictions, Naive Bayes model simply calculates the posterior probability of a new data point by using Bayes Theorem, that can already be found since we have found prior probability earlier.

What is the model's final F1 score?

0.809