Building a Student Intervention System

by Himanshu Panwar , Udacity Machine Learning Nanodegree , PROJECT 2

Report Structure

- Classification vs Regression

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?

This is classification problem, as we have to predict how likely a student is to pass their high
school final exam. So target labels: pass or fail , clearly classification problem.

- Exploring the data

1.Total number of students ?

no. of students : 395

2.Number of students who passed?

passed students : 265

3.Number of students who failed?

failed students : 130

4.Graduation rate of the class?

grad. rate : 67.09%

5.Number of features?

no. of features : 30

- Preparing the Data

1.Identify the feature and target columns, pre-process feature columns , split into training and testing data.

```
Feature column(s):-
['school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Medu', 'Fedu', 'Mjob', 'Fjob', 'reason', 'guardian', 'travelti
me', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', '
famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 'absences']
Target column: passed

Processed feature columns (48):-
['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_r
eputation', 'guardian_father', 'guardian_mother', 'guardian_other', 'traveltime', 'studytime', 'failures', 'schoolsup', '
famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc'
, 'health', 'absences']
```

- Training and Evaluating Models

Choose 3 supervised learning models that are available in scikit-learn, and appropriate for this problem. For each model:

- What are the general applications of this model? What are its strengths and weaknesses?
- Given what you know about the data so far, why did you choose this model to apply?
- Fit this model to the training data, try to predict labels (for both training and test sets), and measure the F1 score. Repeat this process with different training set sizes (100, 200, 300), keeping test set constant.
- Produce a table showing training time, prediction time, F1 score on training set and F1 score on test set, for each training set size.

Learning Model Used:

1.Gaussian Naive Bayes
2.Decision Tree Classifier
3.Support vector machine

NAIVE BAYES:
Based on applying bayes theorem with assumption of independence between every pair of features.
Pros: learning speed ,simplicity
Cons: poor performance if independence assumptions do not hold,zero probability problem.
Applications : spam detection, document classification
This model is chosen for its simplicity and learning speed.

	Training	Size	
	100	200	300
Training Time	0.002	0.002	0.002
Prediction Time	0.001	0.001	0.001
F1 score for training set	0.40	0.81	0.78
F1 score for test set	0.39	0.75	0.79

DECISION TREE CLASSIFIER
Pros:
The cost of predicting data is logarithmic in the number of data points used to train the tree.
Able to handle both numerical and categorical data. Performs well with large datasets.Uses white box model(condition can be explained by boolean logic).
Cons: May overfit data
Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.
Applications: different data mining application.
This model was chosen to get white box model, as majority of the features are mutually exclusive.

	Training	Size	
	100	200	300
Training Time	0.002	0.002	0.005
Prediction Time	0.000	0.000	0.001
F1 score for training set	0.975	0.961	0.961

F1 score for test set	0.666	0.639	0.794

Clearly overfitting the data.
SUPPORT VECTOR MACHINE
Pros:
Effective in cases where number of dimensions is greater than the number of samples.
Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
Custom Kernels
Cons:
If the number of features is much greater than the number of samples, the method is likely to give poor performances.
Picking/finding the right kernel can be a challenge.
Applications: image recognition , text-classification
This model was chosen as features set length 30 , non-linear set of boundaries , use of kernels.

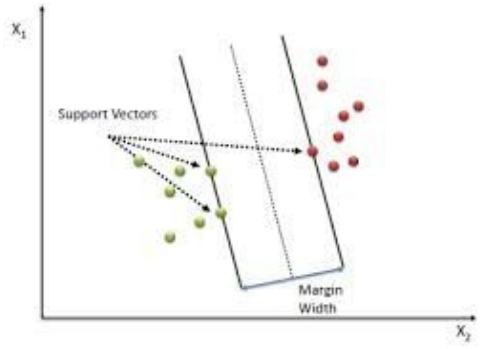
	Training	Size	
	100	200	300
Training Time	0.003	0.006	0.009
Prediction Time	0.002	0.005	0.007
F1 score for training set	0.882	0.881	0.845
F1 score for test set	0.774	0.783	0.833

- Choosing the best model

As per the above table data.
The SVM model gives best F1 score as compared to other models. The least training time is for naive bayes but poor performance as compared to others.
Although SVM has costly time complexity (O(n^3)) but it gives us high performance and can handle large datasets .

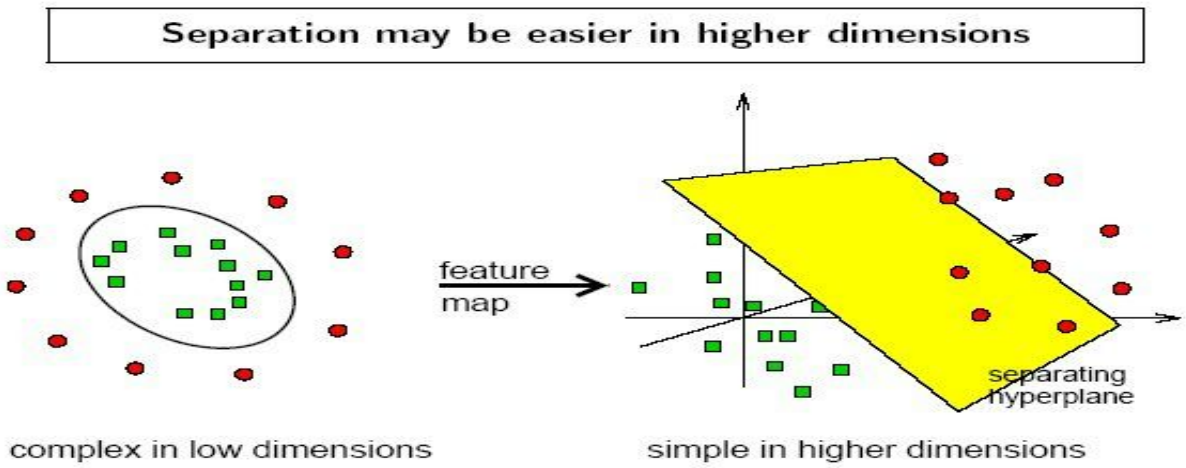
SVM are based on decision lines concept.
A decision line : the one that separates between different classes. Given a labeled dataset in supervised learning, the decision line separates the categorized objects .
A schematic example is shown in the illustration below.

In this example, the objects belong either to class GREEN or RED. The separating line defines a boundary on the left side of which all objects are GREEN and to the right of which all objects are RED. Any new object (white circle) falling to the right is labeled, i.e., classified, as RED (or classified as GREEN should it fall to the left of the separating line).
SVM chooses the best decision line or divide where the distance between that line and the nearest observations of differing classes are the largest (i.e. achieving the largest margin).



The above is a classic example of a linear classifier.

Most classification tasks, however, are not that simple, and often more complex structures are needed in order to make an optimal separation, i.e., correctly classify new objects (test cases) on the basis of the examples that are available (train cases).
This situation is depicted in the illustration below. Compared to the previous schematic, it is clear that a full separation of the GREEN and RED objects would require a curve (which is more complex than a line).
Classification tasks based on drawing separating lines to distinguish between objects of different classes are known as hyperplane classifiers. Support Vector Machines are particularly suited to handle such tasks.



The illustration above shows the basic idea behind Support Vector Machines. Here we see the original objects (left side of the schematic) mapped, i.e., rearranged, using a set of mathematical functions, known as kernels. The process of rearranging the objects is known as mapping (transformation). Note that in this new setting, the mapped objects (right side of the schematic) is linearly separable and, thus, instead of constructing the complex curve (left schematic), all we have to do is to find an optimal line that can separate the GREEN and the RED objects.

Important parameters for the model using Grid-Search
Checked three parameters: C,Gamma,Kernel

```
parameters=  {'gamma': [1e-1, 1e-2, 1e-4, 1e-6],
          'C': [10, 100, 200, 300, 500, 700],
          'kernel':['rbf','sigmoid','poly']
          }
```

Result:

F1 score: 0.864864864865
Best params: {'kernel': 'rbf', 'C': 300, 'gamma': 0.0001}

F1 score on test data : 0.864 better than default SVM model.