## Project 2: Supervised Learning

Building a Student Intervention System

## 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?**

This problem is categorized as classification. The deciding factor to decide this is that the prediction result will be discrete (‘yes’ or ’no’).

## Exploring the Data

**Can you find out the following facts about the dataset?**

* Total number of students: 395
* Number of students who passed: 265
* Number of students who failed: 130
* Graduation rate of the class (%): 67.09 %
* Number of features (excluding the label/target column): 30

## Preparing the Data

## Training and Evaluating Models

**Choose 3 supervised learning models that are available in scikit-learn, and appropriate for this problem.**

Decision Tree, SVM and Naive Bayes.

**What are the general applications of each model? What are their strengths and weaknesses?**

* Decision Tree:
  + **General Applications:**

Decision trees are used mainly for classification problems such as credit risk analysis, medical diagnosis, industrial applications and etc.

* + **Strengths:**

Simple to understand and explain.

Require little data preparation.

Able to handle numerical and also categorical data.

* + **Weaknesses:**

Creating over-complex trees, mostly from overfitting training data or trying to solve non-lineraly separable problems.

Suffer when working with continuous variables.

* SVM:
  + **General Applications**

SVMs can be used to solve classification and regression problems such as image classification, handwritting recognition, stock prediction, etc.

* + **Strengths**

Works very well with smaller datasets.

Can handle a high number of features.

Does not rely on the entire training set to create the boundaries.

Highly accurate.

* + **Weaknesses**

Doesn’t perform well with large datasets

Computationally intensive when training and predicting.

Doesn’t perform well when there’s a lot of noise in the data.

* Naïve Bayes:
  + **General Applications**

Used in the real world with many applications, such as spam detection, sentiment analysis, recommendation systems, customer segmentation, computer vision, etc.

* + **Strengths**

Highly scalable but can be used with a small amount of training data.

It is computationally light-weight.

It performs well with categorical and numerical features.

* + **Weaknesses**

Assumes independent predictors, that means when the data is highly correlated the predictions will suffer.

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

All chosen models benefit from the data size. Despite a large number of features…

**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.**

* Decision Tree:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training set size | | |
| 100 | 200 | 300 |
| Training time (s) | 0.001 | 0.001 | 0.003 |
| Prediction time (s) | 0.000 | 0.000 | 0.000 |
| F1 score for training set | 1 | 1 | 1 |
| F1 score for test set | 0.645 | 0.725 | 0.683 |

* SVM:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training set size | | |
| 100 | 200 | 300 |
| Training time (s) | 0.003 | 0.005 | 0.007 |
| Prediction time (s) | 0.001 | 0.001 | 0.002 |
| F1 score for training set | 0.870 | 0.849 | 0.870 |
| F1 score for test set | 0.80 | 0.80 | 0.80 |

* Naïve Bayes:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Training set size | | |
| 100 | 200 | 300 |
| Training time (s) | 0.002 | 0.001 | 0.001 |
| Prediction time (s) | 0.001 | 0.000 | 0.000 |
| F1 score for training set | 0.796 | 0.814 | 0.819 |
| F1 score for test set | 0.638 | 0.734 | 0.75 |

## 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.**

Based on the simple tests made in this project the model with the highest F1 score is SVM. But SVM is the model with the lowest time efficiency, this the probably due to the large number of features in the dataset. Considering limited resources, cost and performance I chose the decision tree model.

The decision tree model gives a perfect F1 score in the training test. This score probably comes from overfitting the tree to the training data. So, when this model is optimized the F1 score will increase considerably regarding the test data. Also, decision trees have a better performance Naïve Bayes when working with small datasets.

**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).**

Decision trees work by taking data from past students (age, gender, family, study time, absence, etc), asking questions about this data and dividing the answers like branches from a tree. Each division separate the answers from a specific characteristic, for example:

Is the student male of female? If he is male then it goes to the male branch, otherwise it goes to the female branch.

The tree is complete when the information from all past students is used. At the end of each branch in the completed tree there’s an answer telling if a student that followed that branch graduated or not.

With the complete tree you can follow a specific branch using a student’s characteristics to find if he’s more likely to graduate or not.

**Fine-tune the model. What’s the model final F1 score?**

0.774193548387