Project 2: Supervised Learning

Building a Student Intervention System

1. Classification vs Regression

Which type of supervised machine learning problem is this, classification or regression? Why?

This is clearly a classification task. There is a discrete set of outputs, specifically binary, *pass/fail*. There are a large variety of classification algorithms. Three will be explored in this report. If the output, or response, variable came from a continuous set of values, then it would be appropriate to use regression.

2. Exploring the Data

Now, 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

Number of features: 30

Graduation rate of the class: 67.09%

3. Preparing the Data

The target column is the *passed* column. All other are the feature columns.

4. 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?**

First: SVM- Support Vector Machine

What are the general applications of this model?

Classification.

What are its strengths and weaknesses?

Strengths:

Solution to SVMs are global and do not suffer from local minima.Since SVM's are based only on a subset of the training observations, it can be quite robust to observations that lie far away from the separating hyper-plane.When classes are well separated, SVMs perform better than logistic regression, and vice versa.

Weaknesses:

Time Complexity:

> O(n2) according to Scikit Learn SVC documentation. According to research O(n2) for linear kernel. According to research O(n3) for RBF.The n is the number of observations.

Space Complexity:

> O(1).

The parameters of the model are difficult to interpret. The probabilities for class membership are un-calibrated, contributing to the lack of interpretability. There are a large amount of parameters to optimize, and, thus, can take a long time top optimize. Due to their flexibility, kernel models are very sensitive to overfitting.

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

SVMs are purported as excellent out of the box classifiers, meaning they are generally robust and perform well on a variety of data.

Second: AdaBoost- Adaptive Boosting

General Application:

Classification.

AdaBoost is a boosting algorithm that uses ‘weak learners’ (other machine learning algorithms that will perform even slightly better than random classification) to classify data.It works by creating a new vector/column with equal weights for every training example and normalizing the weights. Then it iteratively classifies the data in the training set using the weak learner, determines which examples it got wrong, gives those examples higher weights, and finally renormalizes the weights so that they sum to one. This process will continue until the specified number of estimators has been reached, or classification error has reached zero.

What are its strengths and weaknesses?

Strengths:

In the circumstance of using decision trees, we can stave off overfitting by using a shallower depth than would be used for a single decision tree. AdaBoost generally works well on both training and test sets.It is widely held as one of the top performing meta-algorithms in machine learning.

Weaknesses:

Time Complexity:

O(Nweak learners*O()weak learner) : Decision Tree O(n*depth)

Space Complexity:

O(Nweak learners*O()weak learner): Decision Tree O(n*depth) Learns slowly.

Can overfit if the number of estimators is too large.Storing multiple decision trees and alpha values. It can overfit data if the data is not ‘well-behaved’ aka if it is uniformly random. Boosting algorithms normally have a lot of parameters to tune in order to get the best model.

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

AdaBoost is a robust algorithm that works very well in classification tasks. Given that the data set is small, and the learning speed of the AdaBoost algorithm, I thought it would be appropriate to still use a highly accurate model.

Third: Logistic Regression

What are the general applications of this model?

Classification.

Logistic regression is recommended as one of the first machine learning algorithms to use when addressing a classification task. It is a simple linear model and can be extended and penalized in a variety of ways.

What are its strengths and weaknesses?

Strengths:

Time Complexity:

> Can be as little as O(d) where d is the number of dimensions.

Space Complexity:

> O(1) because you simply store an equation.

Results are very interpretable as the probability of a certain class.The importance of individual features can be interpreted from the coefficients applied to them in the final model.It is also pretty robust to noise and you can avoid overfitting and even do feature selection by using l2 or l1 regularization.Not rigidly tied to feature independence assumption like that of Naïve Bayes.

Weaknesses:

Simply applied Logistic Regression does not take into consideration that some features may be correlated with others, which reduces the meaningfulness of coefficients. This reduces redundancy in its coefficients but can also ignore confounding features. It may not work as well as other learning algorithms for multi-class classification and does not work as well as SVMs when classes are well separated. The coefficients may be unstable.

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

This model generally works about the same as an SVM but with the added benefit of an interpretable model. While this model does not inherently have the flexibility of an SVM with a kernel, it is still robust to observations that lie far away from its linear decision boundary.

	Training Set Size		
Support Vec. Classifier	100	200	300
Training Time (secs)	0.013216	0.005578	0.032964
Prediction Time (secs)	0.001935	0.003049	0.002975
F1 Score for Training Set	0.886076	0.857143	0.868966
F1 Score for Test Set	0.874251	0.865854	0.84472

	Training Set Size		
AdaBoost Classifier	100	200	300
Training Time (secs)	0.503293	0.343045	0.624276
Prediction Time (secs)	0.021856	0.023981	0.015616
F1 Score for Training Set	0.992908	0.862319	0.850746
F1 Score for Test Set	0.760563	0.813793	0.780822

	Training Set Size		
Log. Reg. Classifier	100	200	300
Training Time (secs)	0.002251	0.024206	0.025705
Prediction Time (secs)	0.00029	0.000296	0.000372
F1 Score for Training Set	0.905405	0.836237	0.815534
F1 Score for Test Set	0.744526	0.821192	0.794521

5. Choosing the Best Model

Based on the experiments you performed earlier, in 1-2 paragraphs explain to the board of supervisors what single model you chose as the best model. Which model is generally the most appropriate based on the available data, limited resources, cost, and performance?

Given that for each model the training and predicting times are exceedingly low, combined with the fact that you would not need to run this model dynamically, multiple times a day or week, it is most appropriate to go with the model that has the best F_1 score. The best performing algorithm here is the Support Vector Classifier, and it is the algorithm I have chosen to tune for the final model. However, if you intended to run this model more dynamically, it performs almost the same on 200 observations as it does on 300 observations when considering the F_1 score.

In 1-2 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 make a prediction).

The goal of the SVM is to find a hyper-plane that best distinguishes passing students from failing students. Simply put, if passing students can be divided by drawing a straight barrier between characteristics of those students, then a SVM will find that barrier that separates them. What makes the barrier of a SVM special is that the line maximizes the distinction of characteristics between passing and failing students. In addition, if there are exceptional cases of students passing or failing, the SVM will not be distracted and will focus on distinguishing most of the students correctly. As always, though, we must remember that all algorithm and statistics are subject to the quality of the data we collect. Even so, the SVM will try to find the most robust solution to define which students merit closer attention.

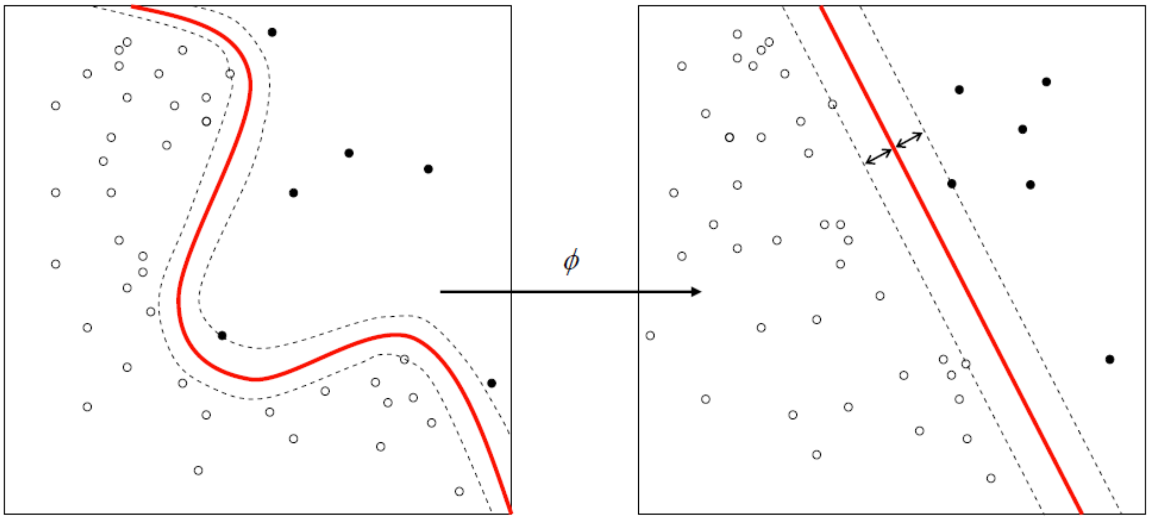

Figure 1) Image provided by Wikipedia: https://en.wikipedia.org/wiki/Support_vector_machine

If the boundary that separates passing from failing students looks like the left part of figure 1, meaning in cannot be separated by simply drawing a line between the passing and failing students, then an SVM can use a kernel function to transform the data into new features that make the barrier look more like the right part of figure 1.

What is the model's final F_1 score?

0.84472049689440998