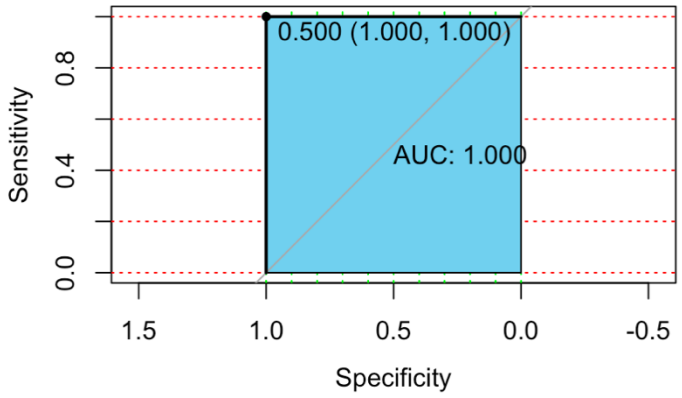
# Biostat 626 Midterm 1: Problem Sets

1. See Canvas.
2. https://github.com/zyixuanUM/626-Midterm
3. https://github.com/zyixuanUM/626-Midterm/blob/main/README.md

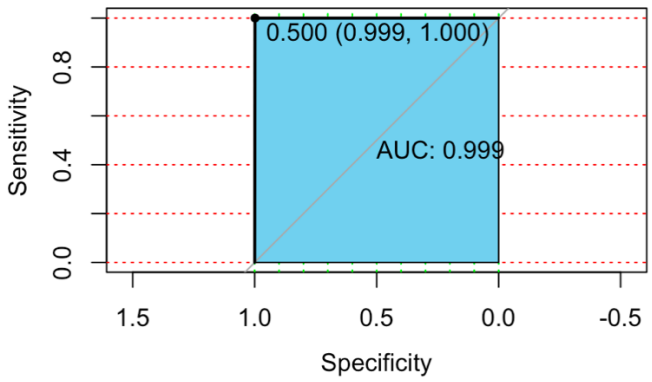
4/5. Binary Classifier:

| Model | Accuracy |
|---|---|
| Logistic | 100% |
| Elastic Net | 99.01% |
| Lasso | 100% |
| Ridge | 99.91% |
| LDA | 100% |
| SVM(linear) | 100% |
| SVM(radial) | 100% |
| Neural Network | 100% |
| Adaboost | 99.91% |

Example Figures:




(SVM)

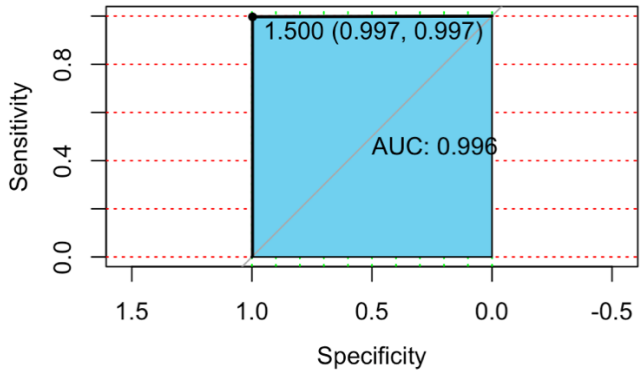



(Adaboost)

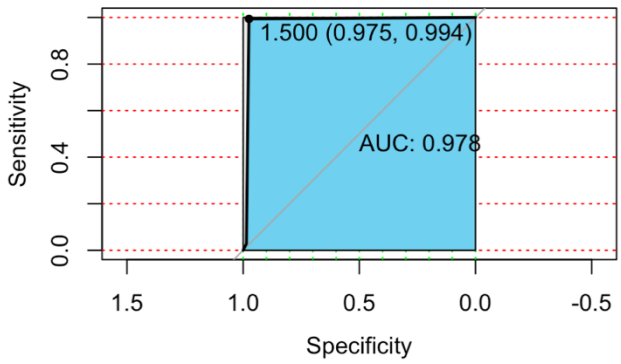
Multi-class Classifier:

| Model | Accuracy |
|---|---|
| Bagging | 88.37% |
| Adaboost | 96.78% |
| SVM(linear) | 98.54% |
| SVM(radial) | 97.47% |
| Randomforest | 98.07% |
| Neural Network | 96.14% |
| LDA | 98.07% |

Example figures:




(SVM)




(Adaboost)

Specific algorithms:

In this project's classifier, we use several methods to build classifiers, such as logistic regression, GLM with elastic net, lasso regression, ridge regression, linear discriminant analysis, SVM with linear kernel or radial kernel, neural network, adaboost, bagging, randomforest, etc. Through these models, use different packages and different parameters to adjust the results, in order to get the best answer. One of the methods and their accuracy are as follows. Most of the functions come from the caret package and other required packages. Both choose the SVM as the final algorithm, for it has one of the highest accuracy and the fastest system time.

## 7. Results and future improvement

The result of each classification from the training data can be seen in table() or confusionMatrix().

For the binary classifier, there are several methods that get 100% accuracy, so we don't need to improve it. Choose the SVM with linear kernel as the final algorithm for it has 100% accuracy with the fastest system time.

However, for the multi-class classifier, there is still space for us to improve it. As I missed some opportunities and submitted a wrong submitted file, I don't have any effective accuracy of testing data here. But I can provide some further improvements as follows. Also, choose the SVM with linear kernel as the final algorithm for it has one of the highest accuracies with the fastest system time.

Specifically, we can try the function caretStack(all.models, ...) in the R package "caretEnsemble". This function is used to find a good linear combination of several classification or regression models, using either linear regression, elastic net regression, or greedy optimization.

We can first use the caretList to build a list of models, or directly build the models by train(x, ...), then make a linear regression ensemble by the code caretStack(all.models, method='glm', trControl), or combine with the randomforest like caretStack(all.models, method='rf', trControl).

This method of combining several predictive models via stacking may have better accuracy on the testing data.