UNIVERSITY *of* WASHINGTON

Report of Ensamble Classification

Yana Liu

15^{th} March 2023

Contents

Resumen

In this problem, we developed a classification model using the heterogeneous ensemble approach, which combined the predictions of four different classification algorithms (SVM, Neural Net, Decision Tree, and Random Forest) to improve overall accuracy. We used the "BreastCancer" dataset from the mlbench R package and split it into training and test sets with a 80/20 ratio.

We trained individual models on the training set and then combined their predictions using a random forest meta-model and evaluated the performance of the ensemble model using a confusion matrix and calculated its accuracy. The results showed that the ensemble model correctly classified the majority of the tumors in the test set, achieving an accuracy of around 96-97%.

Overall, the heterogeneous ensemble approach is a useful technique for improving classification performance by combining the strengths of multiple algorithms. However, it is important to keep in mind the limitations of the data and the potential for overfitting when developing such models.

Load data

```
library(mlbench)
data(BreastCancer)
BreastCancer <- na.omit(BreastCancer)
BreastCancer$Id <- NULL
```

Split train data & test data

Set seed for reproducibility, randomly select 80% of the data for training, and the rest 20% fo the data is for testing.

```
set.seed(2)
ind <- sample(2, nrow(BreastCancer), replace = TRUE, prob=c(0.8, 0.2))
traindata=BreastCancer[ind == 1,]
testdata=BreastCancer[ind == 2,]
```

Load packages

Create classifiers

Next, we train individual models using four different classification algorithms: random forest, SVM with radial kernel, neural network, and decision tree. We store the trained models in separate variables.

Here I use SVM, Neural Net, Decision Tree and Random Forest those 4 classifiers.

We then make predictions on the testing set using each of the individual models, and combine the predictions into a data frame. We use this data frame to train a meta-model, which is a random forest classifier in this case.

SVM

the first classifier is svm.

we can see the Accuracy Table of svm below

```
mysvm <- svm(Class ~ ., data=traindata, probability = TRUE)
mysvm.pred <- predict(mysvm, type="prob", newdata=testdata, probability = TRUE)
matrixsvm <- confusionMatrix(data=mysvm.pred, reference = testdata$Class)
matrixsvm$table
```

```
##            Reference
## Prediction  benign malignant
##   benign        87         1
##   malignant      5        55
```

and the graph of the confusion matrix of svm classifier

Neural Net

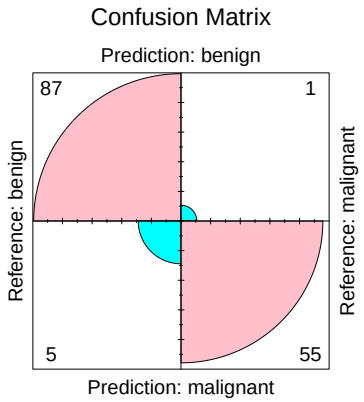

Figure 1: Confusion Matrix of svm

```
mynnet <- nnet(Class ~ ., data=traindata, probability = TRUE, size=1)
```

```
## # weights:  83
## initial  value 452.867181
## iter  10 value 246.343573
## iter  20 value 246.159254
## final  value 246.159242
## converged
```

```
mynnet.pred <- predict(mynnet,type="class", newdata=testdata, probability = TRUE)
mynnet.pred<-as.factor(mynnet.pred)
matrixnnet <- confusionMatrix(data=mynnet.pred, reference = testdata$Class)
```

```
matrixnnet$table
```

```
##            Reference
## Prediction  benign malignant
##   benign        88         6
##   malignant      4        50
```

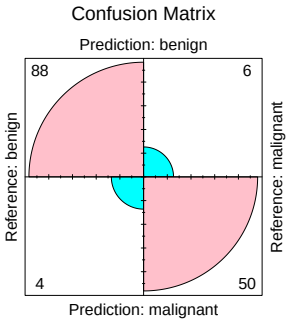

Figure 2: Confusion Matrix of nnet

Decision Tree

```
mytree <- rpart(Class ~ ., data=traindata)
mytree.pred <- predict(mytree,testdata,type="class")
```

```
matrixtree<- confusionMatrix(data=mytree.pred, reference = testdata$Class)
matrixtree$table
```

```
##            Reference
## Prediction  benign malignant
##   benign        86         2
##   malignant      6        54
```

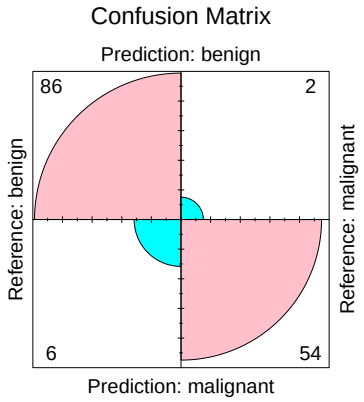

Figure 3: Confusion Matrix of tree

Random Forest

```
myrf <- randomForest(Class~., data = traindata,  probability = TRUE)
myrf.pred <- predict(myrf, type="class", newdata=testdata, probability = TRUE)
matrixrf <- confusionMatrix(data=myrf.pred, reference = testdata$Class)
matrixrf$table
```

```
##            Reference
## Prediction  benign malignant
##   benign        87         0
##   malignant      5        56
```

predict classes for the evaluation data set

Combine the classifiers in an ensemble

Finally, we use the trained ensemble model to make predictions on the combined predictions from the individual models, and evaluate the performance of the model using confusionMatrix function from caret package.

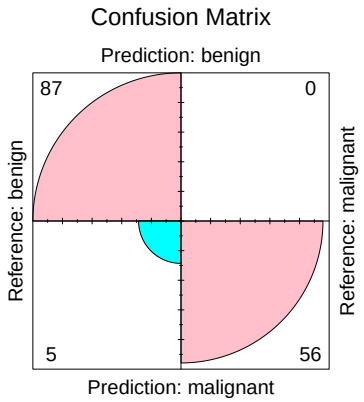

Figure 4: Confusion Matrix of rf

```
mysvm.pred=as.factor(mysvm.pred)
mytree.pred=as.factor(mytree.pred)
mynnet.pred=as.factor(mynnet.pred)
myrf.pred=as.factor(myrf.pred)
ensemblePred <- data.frame(svm = mysvm.pred,nnet=mynnet.pred,tree=mytree.pred,rf = myrf.pred)
```

Train a meta-model on the combined predictions

we need to combine the predicted values from each individual model with the original test data to create a new data frame that includes the true class labels

```
metaModel <- train(Class ~ ., data = cbind(ensemblePred, Class = testdata$Class), method = "rf", trCont
```

Make predictions using the ensemble model

```
ensemblePredFinal <- predict(metaModel, newdata = ensemblePred)
```

Performance Measure

```
confusionMatrix(data = ensemblePredFinal, reference = testdata$Class)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  benign malignant
##   benign        87         0
##   malignant      5        56
##
##                Accuracy : 0.9662
##                  95% CI : (0.9229, 0.9889)
##     No Information Rate : 0.6216
```

```
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9294
##
##  Mcnemar's Test P-Value : 0.07364
##
##             Sensitivity : 0.9457
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9180
##              Prevalence : 0.6216
##          Detection Rate : 0.5878
##    Detection Prevalence : 0.5878
##       Balanced Accuracy : 0.9728
##
##        'Positive' Class : benign
##
```

We can see that the model correctly classified 87 out of 87 benign tumors and 56 out of 61 malignant tumors. However, the model also made 5 false negative predictions for malignant tumors (predicted as benign but actually malignant) and 0 false positive predictions for benign tumors (predicted as malignant but actually benign).

The overall accuracy of the model is 0.9662 or 96.62%. This means that the model correctly classified 96.62% of the tumors in the test set.

Overall, this is a good performance for a classification model, but we need to keep in mind that the sample size of the test set is relatively small, so we should be cautious about drawing general conclusions based on these results. We may need to evaluate the model on a larger dataset or using cross-validation to further validate its performance.