Statistical learning assignment 10

Joshua Damm

2024-05-01

Question a

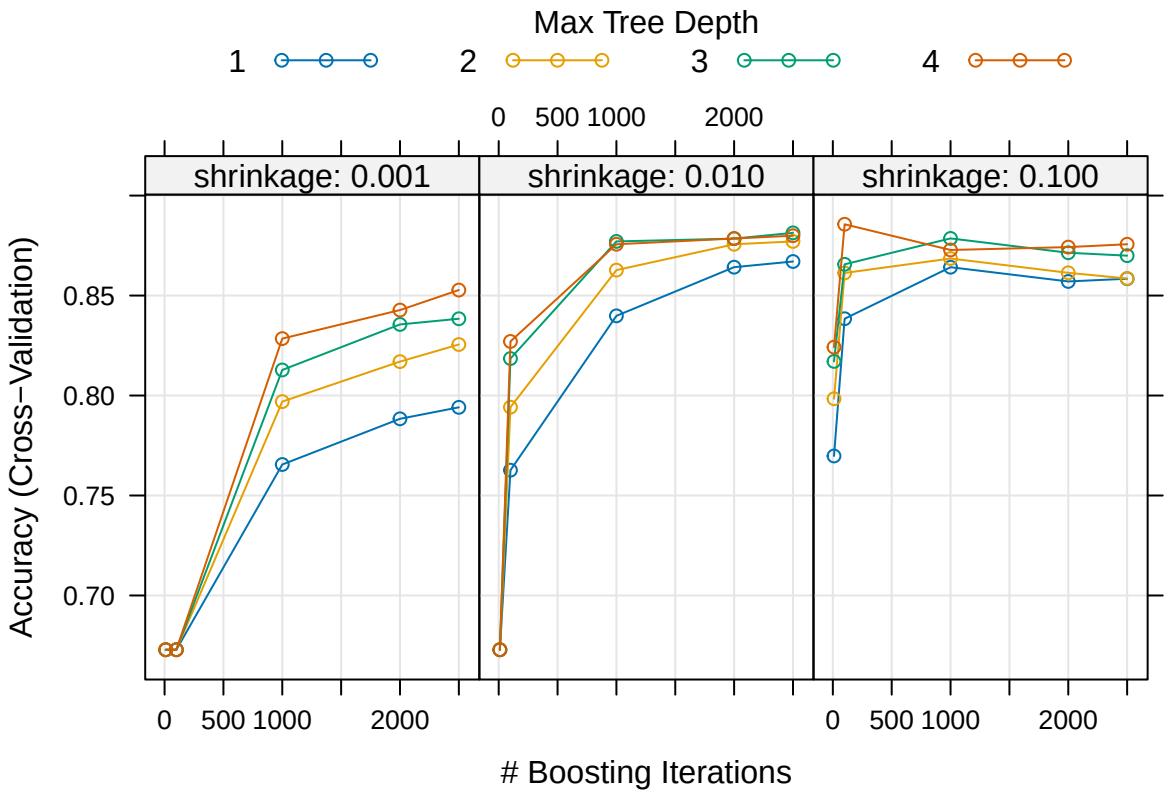
```
# Define the training set proportion
set.seed(4036018)
train <- sample(1:nrow(qsar), size = 700)
test <- which(!(1:nrow(qsar) %in% train))
```

```
# Define the parameter grid
grid <- expand.grid(shrinkage = c(0.1, 0.01, 0.001),
                    n.trees = c(10, 100, 1000, 2000, 2500),
                    interaction.depth = 1:4,
                    n.minobsinnode = 10)
```

Question b

```
## Stochastic Gradient Boosting
##
## 700 samples
##  41 predictor
##   2 classes: 'NRB', 'RB'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 629, 630, 630, 631, 630, 630, ...
## Resampling results across tuning parameters:
##
##   shrinkage  interaction.depth  n.trees  Accuracy   Kappa
##   0.001      1                    10     0.6728644  0.0000000
##   0.001      1                   100     0.6728644  0.0000000
##   0.001      1                  1000     0.7654997  0.4138358
##   0.001      1                  2000     0.7883568  0.4920361
##   0.001      1                  2500     0.7940924  0.5089332
##   0.001      2                    10     0.6728644  0.0000000
##   0.001      2                   100     0.6728644  0.0000000
##   0.001      2                  1000     0.7969691  0.4897442
##   0.001      2                  2000     0.8169702  0.5627006
##   0.001      2                  2500     0.8255215  0.5874049
##   0.001      3                    10     0.6728644  0.0000000
```

```
##   0.001      3                  100     0.6728644  0.0000000
##   0.001      3                 1000     0.8127874  0.5336148
##   0.001      3                 2000     0.8355624  0.6130068
##   0.001      3                 2500     0.8383994  0.6197278
##   0.001      4                   10     0.6728644  0.0000000
##   0.001      4                  100     0.6728644  0.0000000
##   0.001      4                 1000     0.8284816  0.5762628
##   0.001      4                 2000     0.8427466  0.6288943
##   0.001      4                 2500     0.8527472  0.6546010
##   0.010      1                   10     0.6728644  0.0000000
##   0.010      1                  100     0.7626626  0.4060532
##   0.010      1                 1000     0.8398481  0.6317685
##   0.010      1                 2000     0.8641959  0.6870341
##   0.010      1                 2500     0.8670329  0.6935678
##   0.010      2                   10     0.6728644  0.0000000
##   0.010      2                  100     0.7941734  0.4818561
##   0.010      2                 1000     0.8627271  0.6830731
##   0.010      2                 2000     0.8756469  0.7134061
##   0.010      2                 2500     0.8770755  0.7168865
##   0.010      3                   10     0.6728644  0.0000000
##   0.010      3                  100     0.8185017  0.5476398
##   0.010      3                 1000     0.8770755  0.7144714
##   0.010      3                 2000     0.8784839  0.7194869
##   0.010      3                 2500     0.8813618  0.7269961
##   0.010      4                   10     0.6728644  0.0000000
##   0.010      4                  100     0.8270525  0.5747278
##   0.010      4                 1000     0.8756469  0.7125543
##   0.010      4                 2000     0.8785248  0.7206693
##   0.010      4                 2500     0.8799332  0.7238037
##   0.100      1                   10     0.7697653  0.4210152
##   0.100      1                  100     0.8384189  0.6296424
##   0.100      1                 1000     0.8641769  0.6872869
##   0.100      1                 2000     0.8570548  0.6731656
##   0.100      1                 2500     0.8584218  0.6767311
##   0.100      2                   10     0.7983994  0.5000835
##   0.100      2                  100     0.8612985  0.6805052
##   0.100      2                 1000     0.8685254  0.6975667
##   0.100      2                 2000     0.8613819  0.6820943
##   0.100      2                 2500     0.8585455  0.6757399
##   0.100      3                   10     0.8170732  0.5484713
##   0.100      3                  100     0.8656262  0.6872701
##   0.100      3                 1000     0.8785869  0.7205523
##   0.100      3                 2000     0.8714032  0.7036202
##   0.100      3                 2500     0.8699539  0.7009363
##   0.100      4                   10     0.8241947  0.5652297
##   0.100      4                  100     0.8856481  0.7335746
##   0.100      4                 1000     0.8727904  0.7054894
##   0.100      4                 2000     0.8742396  0.7077997
##   0.100      4                 2500     0.8756682  0.7109899
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 100, interaction.depth =
##  4, shrinkage = 0.1 and n.minobsinnode = 10.
```



Question c

Generally, models with a higher shrinkage (0.1) reach higher accuracy sooner, while those with lower shrinkage (0.001) take longer to converge, requiring more iterations. Accuracy improves as the number of trees increases, but this improvement plateaus after a certain number of trees (around 1000-2000 for lower shrinkage rates and fewer iterations for higher rates). Adding more trees enhances model performance, but beyond a certain point, it may not add substantial value and could risk overfitting.Models with higher interaction depth (3 and 4) generally show higher accuracy compared to those with lower depth (1 and 2). This indicates the model's ability to capture more complex relationships. Higher interaction depth can improve model accuracy by accounting for more intricate interactions among features, but deeper models may also risk overfitting. Interestingly, the model with a rather high shrinkage parameter (0.1) converges very quickly and yields by far the highest accuracy, which is a bit surprising since usually low shrinkage parameter yield to a more fine-tuned model and possibly higher accuracy over many iterations.

Question d

As a higher interaction depth, i.e. allowing more branches and terminal nodes in the seperate trees used to fit the model, yields by far the best accuracy, this suggests that there might be a lot of complex interactive relationships between the variables in the data in order to classify ready or not-ready biodegradable molecules.

Question e

```
# Get the best tuning parameters
bestTune <- gbm_model$bestTune

# setup data
gbm_dat <- qsar
gbm_dat$class <- as.numeric(gbm_dat$class) - 1

# Refit the model
set.seed(4036018)
best_gbm <- gbm(class ~ ., data = gbm_dat[train, ],
                distribution = "bernoulli",
                n.trees = bestTune$n.trees,
                interaction.depth = bestTune$interaction.depth,
                shrinkage = bestTune$shrinkage,
                n.minobsinnode = bestTune$n.minobsinnode)
```

Question f

MCR from previous exercise

CART CART_pruned ctree bag rf boosting 0.1690141 0.1802817 0.1718310 0.1211268 0.1154930 0.1436620

As we can see from the results the boosted ensemble performs better than the single tree, the pruned single decision tree. Both the bagged and random forest ensemble and the gradient-boosted ensemble with default settings perform show higher accuracy (lower MCR).