1 Read in the data

2 Cross-validated Random Forest

group5AA_Black-Boopathy_2_RF

Code

1 Read in the data

The following dataset will be used to predict `classification`. Your goal in this analysis is to create a model able to accurately detect and flag malware. We have a data set containing 50000/50000 malware and benign files. The data set was balanced by selecting only the malware files and downsampling the benign files. The response variable is called `classification`.

Hide

```
train=readRDS("group5AA_Black-Boopathy_train.rds")
holdout=readRDS("holdout_df_Singletree.rds")
```

Hide

```
cores=parallel::detectCores()
cl <- parallel::makeCluster(cores-1)  # Set CPU cores for parallel execution
registerDoParallel(cl)  # Register parallel backend
```

2 Cross-validated Random Forest

2.1 Fit the model

- Use `set.seed(123)`. Use the `caret` package and 5-fold cross-validation to fit a random forest model to the **training data**.
- Create a grid that evaluates for mtry from 1 to 10. Use a minimum node-size of 50 and ensemble 30 trees (`ntree=30`).
- Print the model.

This will take a few minutes to run. If you want to speed it up, you can use parallel processing with the train function.

Hide

```
set.seed(123)
# Set up trainControl
cv_control <- trainControl(method = "cv", number = 5,allowParallel = TRUE )

# Define a grid of `mtry` values to search over
mtry_grid <- expand.grid(mtry = seq(1, ncol(train) - 1, by = 1))

# Train the Random Forest model with cross-validation
rf_cv <- train(loan_default ~ .,
                  data = train,
                  method = "rf",  # Uses randomForest under the hood
                  trControl = cv_control,  # Apply 5-fold CV
                  tuneGrid = mtry_grid,  # Search over `mtry`
                  ntree = 30,  # Fixed number of trees
                  importance = TRUE,
                  nodesize = 50)
```

2.2 Variable Importance

Print the variable importance plot for the model

Hide

```
rf_cv
```

```
## Random Forest
##
## 226434 samples
##     34 predictor
##      2 classes: 'No', 'Yes'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 181148, 181148, 181146, 181147, 181147
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    1    0.8078645  3.713126e-05
##    2    0.8193734  1.036532e-01
##    3    0.8218421  1.441971e-01
##    4    0.8215595  1.533495e-01
##    5    0.8215021  1.560532e-01
##    6    0.8217803  1.607412e-01
##    7    0.8213740  1.621321e-01
##    8    0.8210075  1.602119e-01
##    9    0.8214049  1.663139e-01
##   10    0.8211090  1.659773e-01
##   11    0.8210737  1.651548e-01
##   12    0.8213431  1.677905e-01
##   13    0.8208971  1.674350e-01
##   14    0.8207336  1.675479e-01
##   15    0.8207513  1.676715e-01
##   16    0.8204466  1.667640e-01
##   17    0.8209015  1.699633e-01
##   18    0.8207248  1.703035e-01
##   19    0.8202081  1.688675e-01
##   20    0.8205261  1.710536e-01
##   21    0.8201684  1.694362e-01
##   22    0.8204598  1.698157e-01
##   23    0.8201242  1.697145e-01
##   24    0.8199078  1.696017e-01
##   25    0.8198283  1.696477e-01
##   26    0.8198151  1.703075e-01
##   27    0.8206806  1.752402e-01
##   28    0.8199166  1.683238e-01
##   29    0.8202214  1.723790e-01
##   30    0.8200138  1.730297e-01
##   31    0.8203450  1.745852e-01
##   32    0.8200403  1.735232e-01
##   33    0.8199343  1.752988e-01
##   34    0.8194706  1.709161e-01
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

Hide

```
holdout$rf_cv.class <- predict(rf_cv,
                                      newdata=holdout,
                                      type ="raw"
                                      )
holdout$rf_cv.prob <- predict(rf_cv,
                                      newdata=holdout,
                                      type ="prob"
                                      )[,"Yes"] #probability of "Yes"

confusionMatrix(holdout$rf_cv.class,
                holdout$loan_default,positive="Yes"


            )
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    No   Yes
##        No  60566 12962
##        Yes   460  1567
##                                           
##                Accuracy : 0.8224          
##                  95% CI : (0.8196, 0.8251)
##     No Information Rate : 0.8077          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.1492          
##                                           
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##             Sensitivity : 0.10785         
##             Specificity : 0.99246         
##          Pos Pred Value : 0.77306         
##          Neg Pred Value : 0.82371         
##              Prevalence : 0.19230         
##          Detection Rate : 0.02074         
##    Detection Prevalence : 0.02683         
##       Balanced Accuracy : 0.55016         
##                                           
##        'Positive' Class : Yes             
## 
```

Hide

```
stopCluster(cl)  # Shut down parallel cluster
registerDoSEQ()  # Reset to sequential processing
```

Hide

```
saveRDS(holdout, "1holdout_df_RF.rds")
```