Preparation

Run this cell to clear the variables in your global R environment.

```
rm(list = ls())
ls()
```

```
## character(0)
```

Libraries

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(RWeka)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Data files

```
lab1_noVPN = read.csv("slackPcapFixGapLab1_1111_flows.csv")
lab1_1111 = read.csv("slackPcapFixGapLab1_flows.csv")


# Drop Src.ipaddr && Dest.ipaddr && start,end time && mac
drops <- c("srcIP", "srcIPCC", "srcIPOrg", "dstIP", "dstIPCC", "dstIPOrg",
           "timeFirst", "timeLast", "ethVlanID", "srcMac", "dstMac", "srcMac_dstMac_numP",
           "dstPortClass")
lab1_noVPN <- lab1_noVPN[, !(names(lab1_noVPN) %in% drops)]
lab1_1111 <- lab1_1111[, !(names(lab1_1111) %in% drops)]
sum(is.na(lab1_noVPN))
```

```
## [1] 0
```

```
sum(is.na(lab1_1111))
```

```
## [1] 0
```

```
# Add a new column, isVPN, set to 0 for no_vpn, 1 for 1111
lab1_noVPN$isVPN <- 0
lab1_1111$isVPN <- 1

# Show summary
lab1_noVPN = data.frame(lab1_noVPN)
lab1_1111 = data.frame(lab1_1111)
```

Prepare Data

```
# Find length of the no_vpn table
n_noVPN = length(lab1_noVPN$flowInd)

# Find length of the vpn table
n_VPN = length(lab1_1111$flowInd)

if (n_noVPN > n_VPN) {
  nTrain = n_VPN*0.7
} else {
  nTrain = n_noVPN*0.7
}

# Define Training set & Testing set
prop = nTrain/(nrow(lab1_noVPN))
set.seed(123)
trnrows_noVPN  <- sample(nrow(lab1_noVPN),nrow(lab1_noVPN)*prop)
dtrain_noVPN <- lab1_noVPN[ trnrows_noVPN,]
dtest_noVPN  <- lab1_noVPN[-trnrows_noVPN,]

trnrows_1111  <- sample(nrow(lab1_1111),nrow(lab1_1111)*0.7)
dtrain_1111 <- lab1_1111[ trnrows_1111,]
dtest_1111  <- lab1_1111[-trnrows_1111,]

dtrain <- rbind(dtrain_noVPN,dtrain_1111)
dtest <- rbind(dtest_1111,dtest_noVPN)

# Remove all columns with only 1 unique value
dtrain <- dtrain %>% select(where(~ n_distinct(.) > 1))
dtrain$isVPN <- as.factor(dtrain$isVPN)

dtest <- dtest %>% select(where(~ n_distinct(.) > 1))
dtest$isVPN <- as.factor(dtest$isVPN)

#dtest <- dtest[, !(names(dtest) %in% c("tcpSeqFaultCnt"))]

nrow(dtest)
```

```
## [1] 226
```

```
nrow(dtrain)
```

```
## [1] 523
```

Train Model

```
train_control<- trainControl(method="cv", number=10)
C45Fit <- train(isVPN ~., method="J48", data=dtrain,
                tuneLength = 5,
                trControl = train_control)
```

```
# Validation
C45Fit
```

```
## C4.5-like Trees
```

```
##
## 523 samples
##  81 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 471, 470, 471, 471, 470, 470, ...
## Resampling results across tuning parameters:
##
##   C       M  Accuracy  Kappa
##   0.0100  1  1         1
##   0.0100  2  1         1
##   0.0100  3  1         1
##   0.0100  4  1         1
##   0.0100  5  1         1
##   0.1325  1  1         1
##   0.1325  2  1         1
##   0.1325  3  1         1
##   0.1325  4  1         1
##   0.1325  5  1         1
##   0.2550  1  1         1
##   0.2550  2  1         1
##   0.2550  3  1         1
##   0.2550  4  1         1
##   0.2550  5  1         1
##   0.3775  1  1         1
##   0.3775  2  1         1
##   0.3775  3  1         1
##   0.3775  4  1         1
##   0.3775  5  1         1
##   0.5000  1  1         1
##   0.5000  2  1         1
##   0.5000  3  1         1
##   0.5000  4  1         1
##   0.5000  5  1         1
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were C = 0.01 and M = 1.
```

```r
C45Fit$finalModel
```

```
## J48 pruned tree
## ------------------
##
## tcpWS <= 0
## |   tcpAnomaly <= 4097: 0 (110.0)
## |   tcpAnomaly > 4097: 1 (15.0)
## tcpWS > 0: 1 (398.0)
##
## Number of Leaves  :  3
##
## Size of the tree :   5
```

```r
predictions = predict(C45Fit, newdata = dtest)
confusionMatrix(predictions, dtest$isVPN)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1
##          0  48   0
##          1   0 178
##                                      
##                Accuracy : 1          
##                  95% CI : (0.9838, 1)
##     No Information Rate : 0.7876     
##     P-Value [Acc > NIR] : < 2.2e-16  
##                                      
##                   Kappa : 1          
##                                      
##  Mcnemar's Test P-Value : NA         
##                                      
##             Sensitivity : 1.0000     
##             Specificity : 1.0000     
##          Pos Pred Value : 1.0000     
##          Neg Pred Value : 1.0000     
##              Prevalence : 0.2124     
##          Detection Rate : 0.2124     
##    Detection Prevalence : 0.2124     
##       Balanced Accuracy : 1.0000     
##                                      
##        'Positive' Class : 0          
## 
```