1. Importing the data and helper function(s)

```
library(readxl)
data <- read_excel("~/RStudio/CS_859_Team_Project/Dec1/Drug_Consumption_data_decade.x
lsx")

f_cols = c("Gender", "Country", "Ethnicity", "Amphet", "Amyl", "Benzos", "Cannabis",
"Crack", "Coke", "Ecstasy", "Legalh", "LSD", "Nicotine")

#drugs = list("Amphet", "Amyl", "Benzos", "Cannabis", "Coke", "Ecstasy", "Legalh", "L
SD", "Nicotine")

data[f_cols] <- lapply(data[f_cols], factor)

data
```

```
## # A tibble: 1,885 × 32
##       ID     Age Gender   Educa…¹ Country Ethni…² Nscore   Escore  Oscore Ascore
##    <dbl>   <dbl> <fct>      <dbl> <fct>   <fct>    <dbl>    <dbl>   <dbl>  <dbl>
##  1     1  0.498  0.48246  -0.0592 0.96082 0.126    0.313 -0.575   -0.583  -0.917
##  2     2 -0.0785 -0.48246  1.98   0.96082 -0.316… -0.678  1.94     1.44    0.761
##  3     3  0.498  -0.48246 -0.0592 0.96082 -0.316… -0.467  0.805   -0.847  -1.62
##  4     4 -0.952  0.48246   1.16   0.96082 -0.316… -0.149 -0.806   -0.0193  0.590
##  5     5  0.498  0.48246   1.98   0.96082 -0.316…  0.735 -1.63    -0.452  -0.302
##  6     6  2.59   0.48246  -1.23   0.24923 -0.316… -0.678 -0.300   -1.56    2.04
##  7     7  1.09   -0.48246  1.16   -0.570… -0.316… -0.467 -1.09    -0.452  -0.302
##  8     8  0.498  -0.48246 -1.74   0.96082 -0.316… -1.33   1.94    -0.847  -0.302
##  9     9  0.498  0.48246  -0.0592 0.24923 -0.316…  0.630  2.57    -0.976   0.761
## 10    10  1.82   -0.48246  1.16   0.96082 -0.316… -0.246  0.00332 -1.42    0.590
## # … with 1,875 more rows, 22 more variables: Cscore <dbl>, Impulsiveness <dbl>,
## #   SS <dbl>, Alcohol <dbl>, Amphet <fct>, Amyl <fct>, Benzos <fct>,
## #   Caff <dbl>, Cannabis <fct>, Choc <dbl>, Coke <fct>, Crack <fct>,
## #   Ecstasy <fct>, Heroin <dbl>, Ketamine <dbl>, Legalh <fct>, LSD <fct>,
## #   Meth <dbl>, Mushrooms <dbl>, Nicotine <fct>, VSA <dbl>, Label <dbl>, and
## #   abbreviated variable names ¹Education, ²Ethnicity
```

```
amphet = data[c("Age", "Nscore", "Oscore", "Cscore", "Impulsiveness", "SS", "Amphet")
]

amyl = data[c("Nscore", "Oscore", "Cscore", "SS", "Amyl")]

benzos = data[c("Age", "Gender", "Nscore", "Escore", "Impulsiveness", "SS", "Benzos")
]

cannabis = data[c("Age", "Education", "Oscore", "Ascore", "Cscore", "Impulsiveness",
"Cannabis")]

crack = data[c("Escore", "Cscore", "Crack")]

coke = data[c("Age", "Oscore", "Ascore", "Impulsiveness", "SS", "Coke")]

ecstasy = data[c("Age", "Gender", "SS", "Ecstasy")]

legalh = data[c("Age", "Gender", "Oscore", "Ascore", "Cscore", "SS", "Legalh")]

lsd = data[c("Age", "Gender", "Nscore", "Escore", "Oscore", "Impulsiveness", "LSD")]

nicotine = data[c("Gender", "Nscore", "Escore", "Cscore", "Nicotine")]
```

```
f1 <- function(rf_model)

{
p = rf_model$confusion[4]/(rf_model$confusion[3]+rf_model$confusion[4])
r = rf_model$confusion[4]/(rf_model$confusion[2]+rf_model$confusion[4])
#sp = rf_model$confusion[4]/(rf_model$confusion[2]+rf_model$confusion[4])
#se = rf_model$confusion[1]/(rf_model$confusion[2]+rf_model$confusion[4])

score = (2*p*r)/(r+p)

#sprintf('Sensitivity: %.4f', se)
#sprintf('Specificity: %.4f', sp)
sprintf('Precision: %.4f, Recall: %.4f, F1 Score: %.4f',  p, r,score)
}
```

2. Running a Random Forest

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

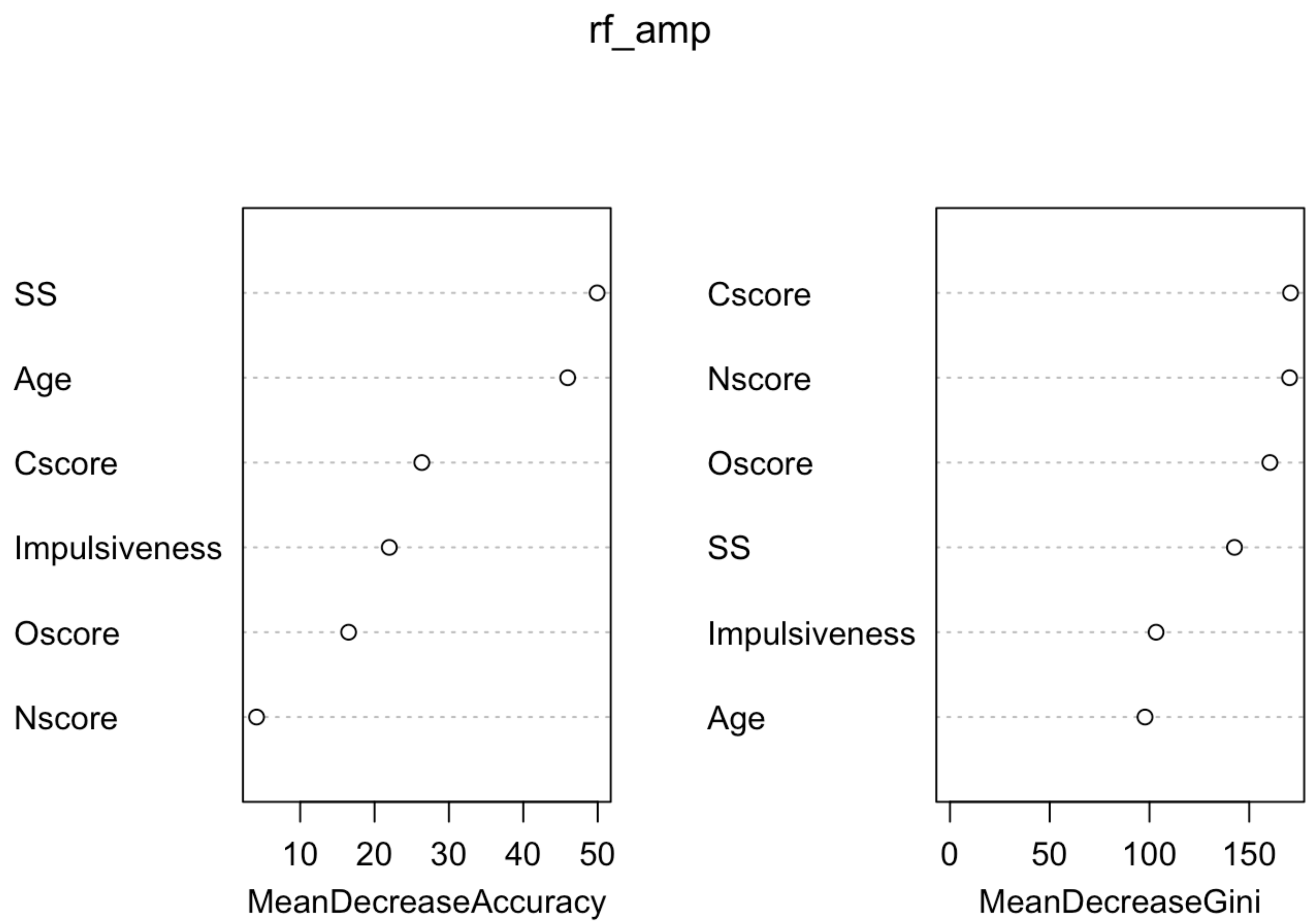
```r
set.seed(365)

rf_amp <- randomForest(Amphet~.,
                    data = amphet,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_amp)
```

```
##
## Call:
##  randomForest(formula = Amphet ~ ., data = amphet, importance = TRUE,      mtry =
2, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 32.1%
## Confusion matrix:
##     0   1 class.error
## 0 942 264   0.2189055
## 1 341 338   0.5022091
```

```r
f1(rf_amp) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.5615, Recall: 0.4978, F1 Score: 0.5277"
```

```r
varImpPlot(rf_amp)
```



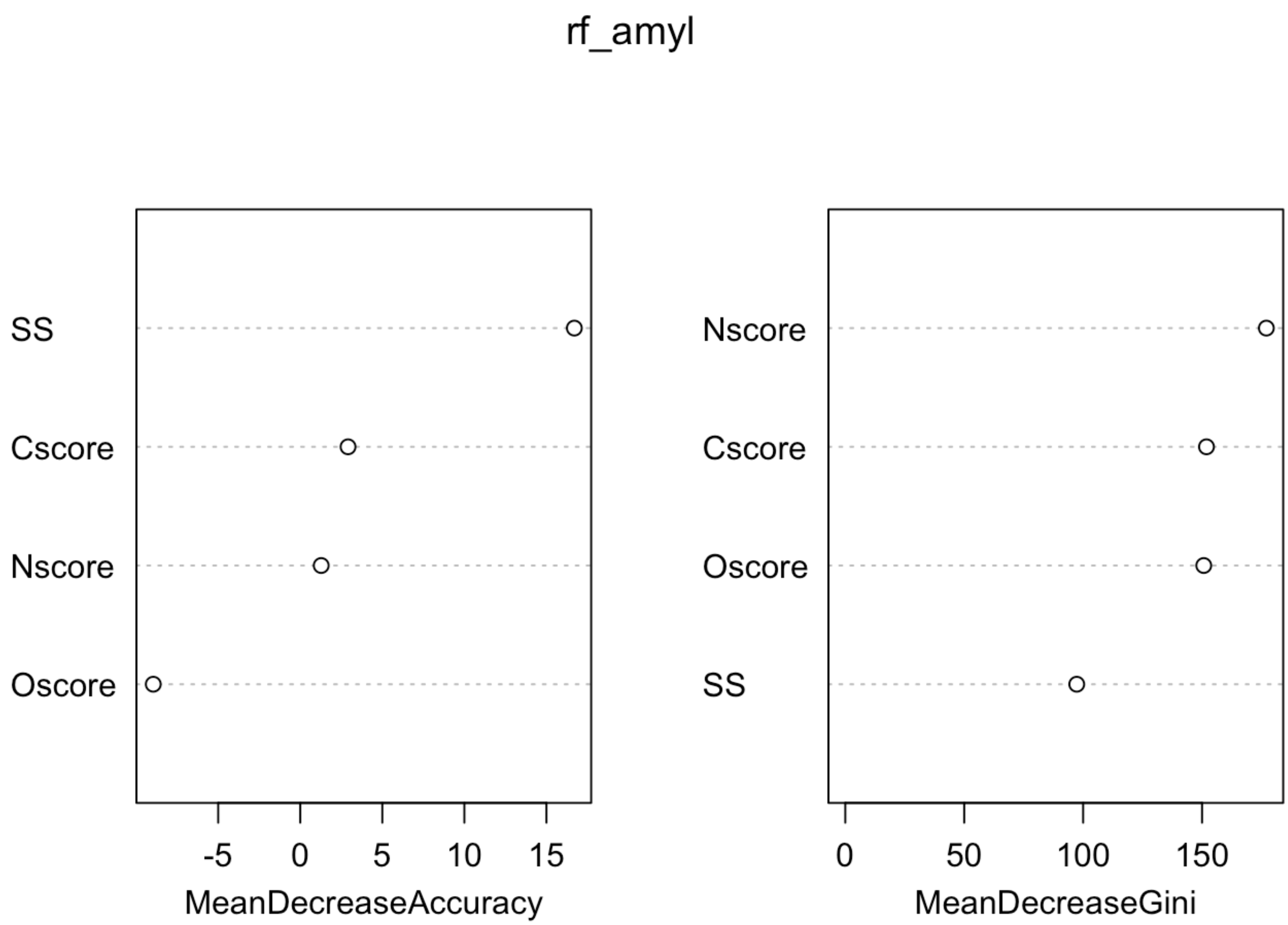

```
set.seed(365)

rf_amyl <- randomForest(Amyl~.,
                    data = amyl,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_amyl)
```

```
##
## Call:
##  randomForest(formula = Amyl ~ ., data = amyl, importance = TRUE,      mtry = 2, n
tree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 22.55%
## Confusion matrix:
##      0  1 class.error
## 0 1436 79  0.05214521
## 1  346 24  0.93513514
```

```
f1(rf_amyl) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.2330, Recall: 0.0649, F1 Score: 0.1015"
```

```
varImpPlot(rf_amyl)
```



```r
set.seed(365)

rf_ben <- randomForest(Benzos~.,
                    data = benzos,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_ben)
```

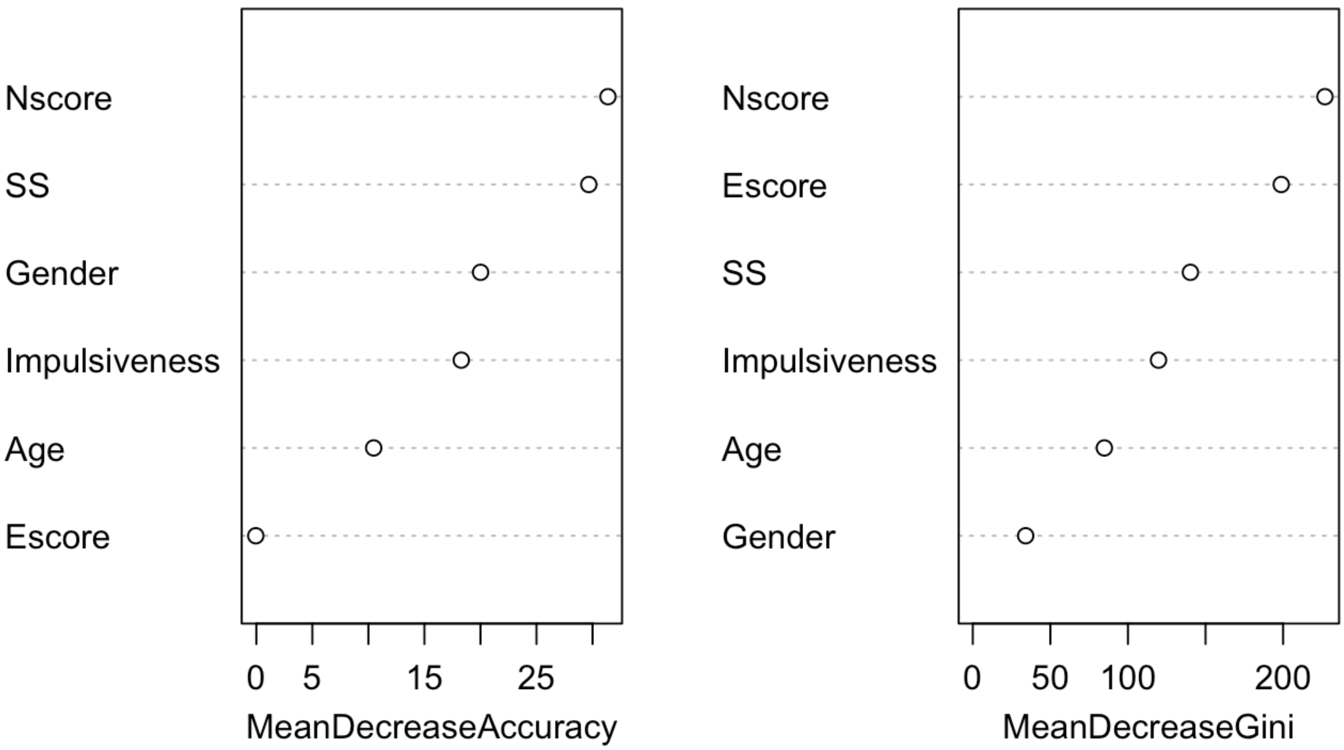
```
##
## Call:
##  randomForest(formula = Benzos ~ ., data = benzos, importance = TRUE,      mtry = 
2, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 35.49%
## Confusion matrix:
##     0   1 class.error
## 0 849 267   0.2392473
## 1 402 367   0.5227568
```

```r
f1(rf_ben) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.5789, Recall: 0.4772, F1 Score: 0.5232"
```

```r
varImpPlot(rf_ben)
```

rf_ben



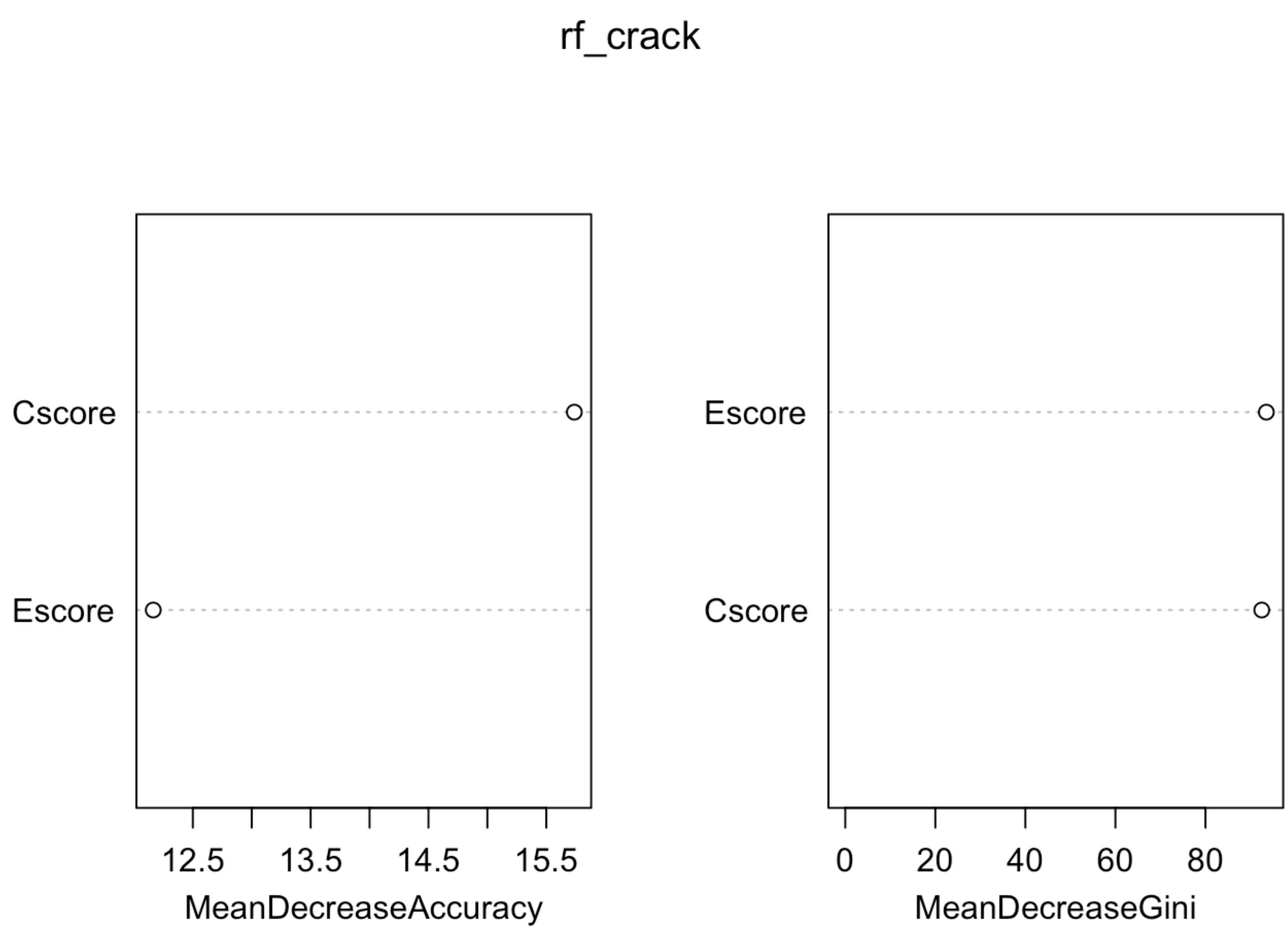

```
set.seed(365)

rf_crack <- randomForest(Crack~.,
                    data = crack,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_crack)
```

```
##
## Call:
##  randomForest(formula = Crack ~ ., data = crack, importance = TRUE,      mtry = 2,
ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 12.63%
## Confusion matrix:
##      0  1 class.error
## 0 1644 50  0.02951594
## 1  188  3  0.98429319
```

```
f1(rf_crack) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.0566, Recall: 0.0157, F1 Score: 0.0246"
```

```
varImpPlot(rf_crack)
```



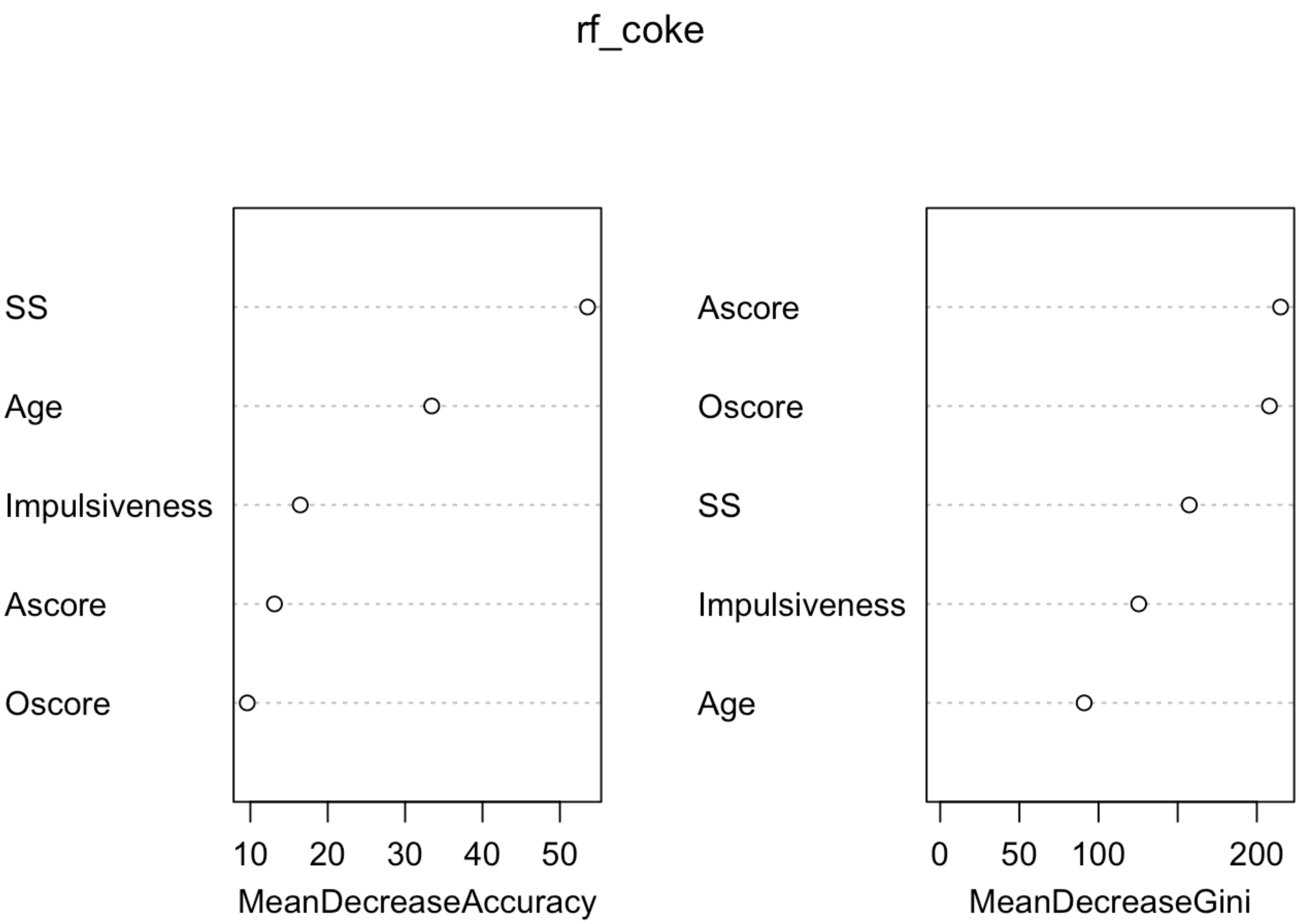
```
set.seed(365)

rf_coke <- randomForest(Coke~.,
                    data = coke,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_coke)
```

```
##
## Call:
##  randomForest(formula = Coke ~ ., data = coke, importance = TRUE,      mtry = 2, n
tree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 35.97%
## Confusion matrix:
##     0   1 class.error
## 0 924 274   0.2287145
## 1 404 283   0.5880640
```

```
f1(rf_coke) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.5081, Recall: 0.4119, F1 Score: 0.4550"
```

```
varImpPlot(rf_coke)
```



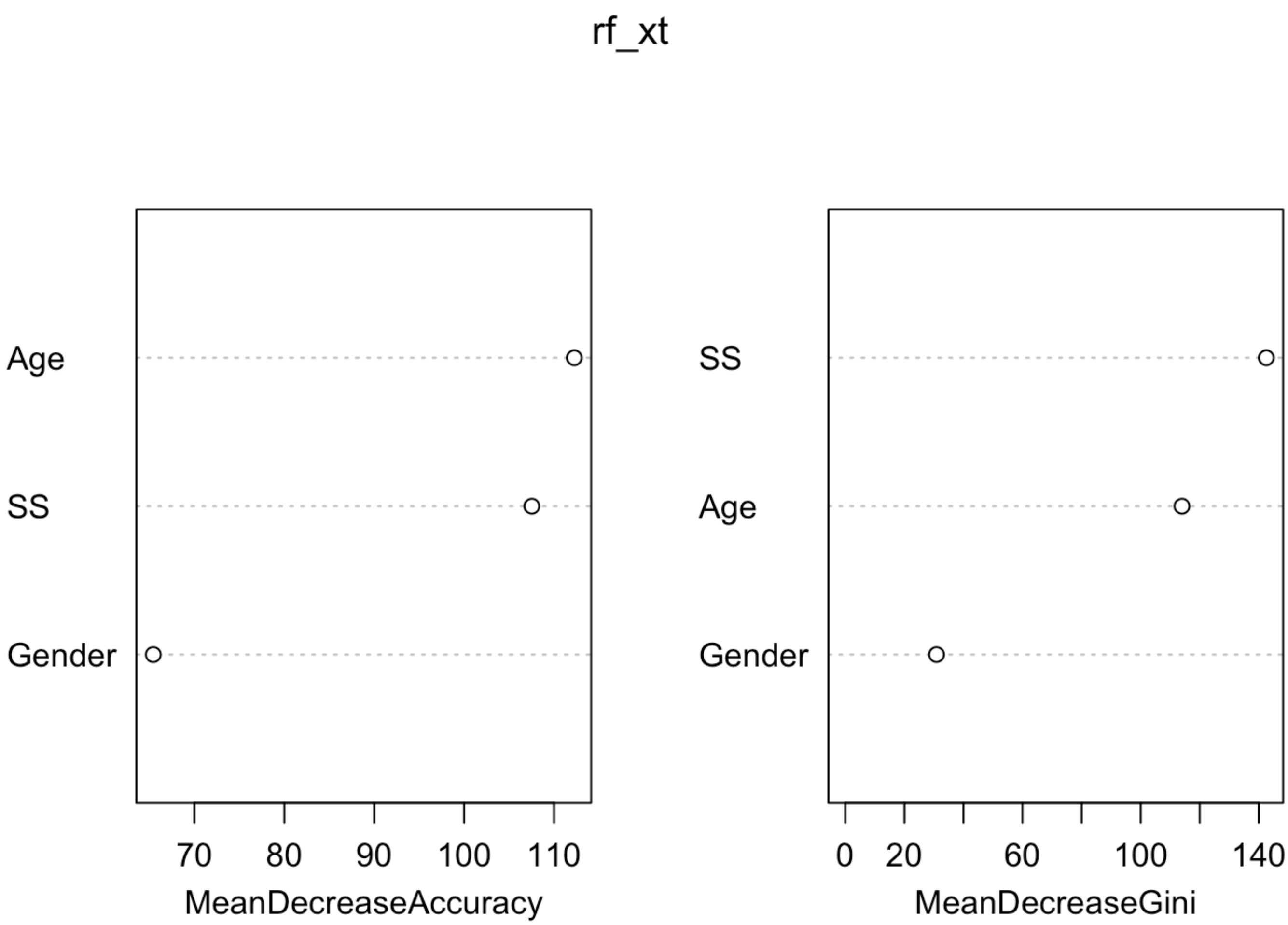

```
set.seed(365)

rf_xt <- randomForest(Ecstasy~.,
                      data = ecstasy,
                      importance = TRUE,
                      mtry = 2,
                      ntree = 1000,
                      CUTOFF = .6,
                      verbose = TRUE)
print(rf_xt)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy, importance = TRUE,      mtry
= 2, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 27.11%
## Confusion matrix:
##     0   1 class.error
## 0 872 262   0.2310406
## 1 249 502   0.3315579
```

```
f1(rf_xt) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6571, Recall: 0.6684, F1 Score: 0.6627"
```

```
varImpPlot(rf_xt)
```



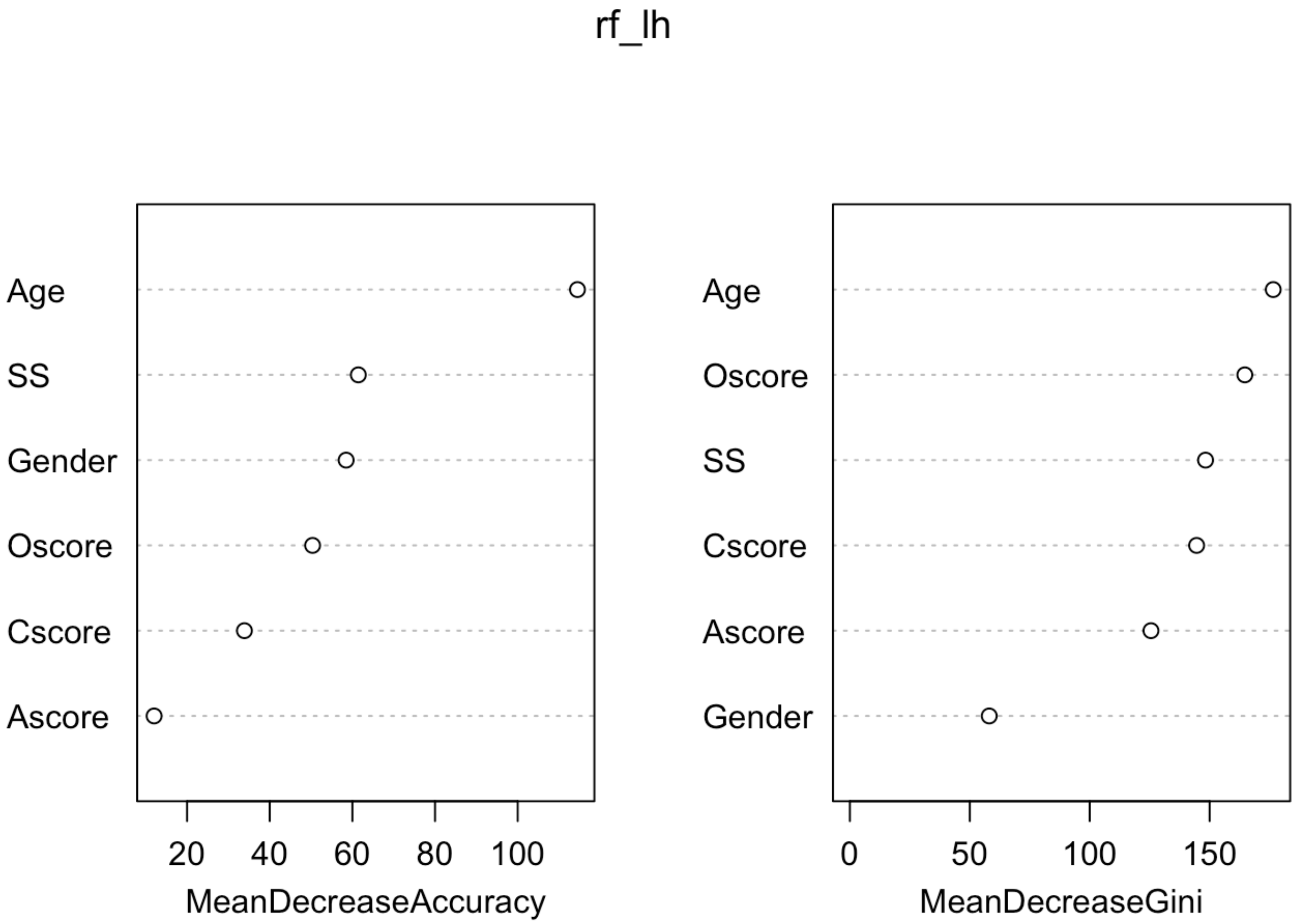
```
set.seed(365)

rf_lh <- randomForest(Legalh~.,
                    data = legalh,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_lh)
```

```
##
## Call:
##  randomForest(formula = Legalh ~ ., data = legalh, importance = TRUE,      mtry =
2, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 21.27%
## Confusion matrix:
##     0   1 class.error
## 0 939 184   0.1638468
## 1 217 545   0.2847769
```

```
f1(rf_lh) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.7476, Recall: 0.7152, F1 Score: 0.7311"
```

```
varImpPlot(rf_lh)
```



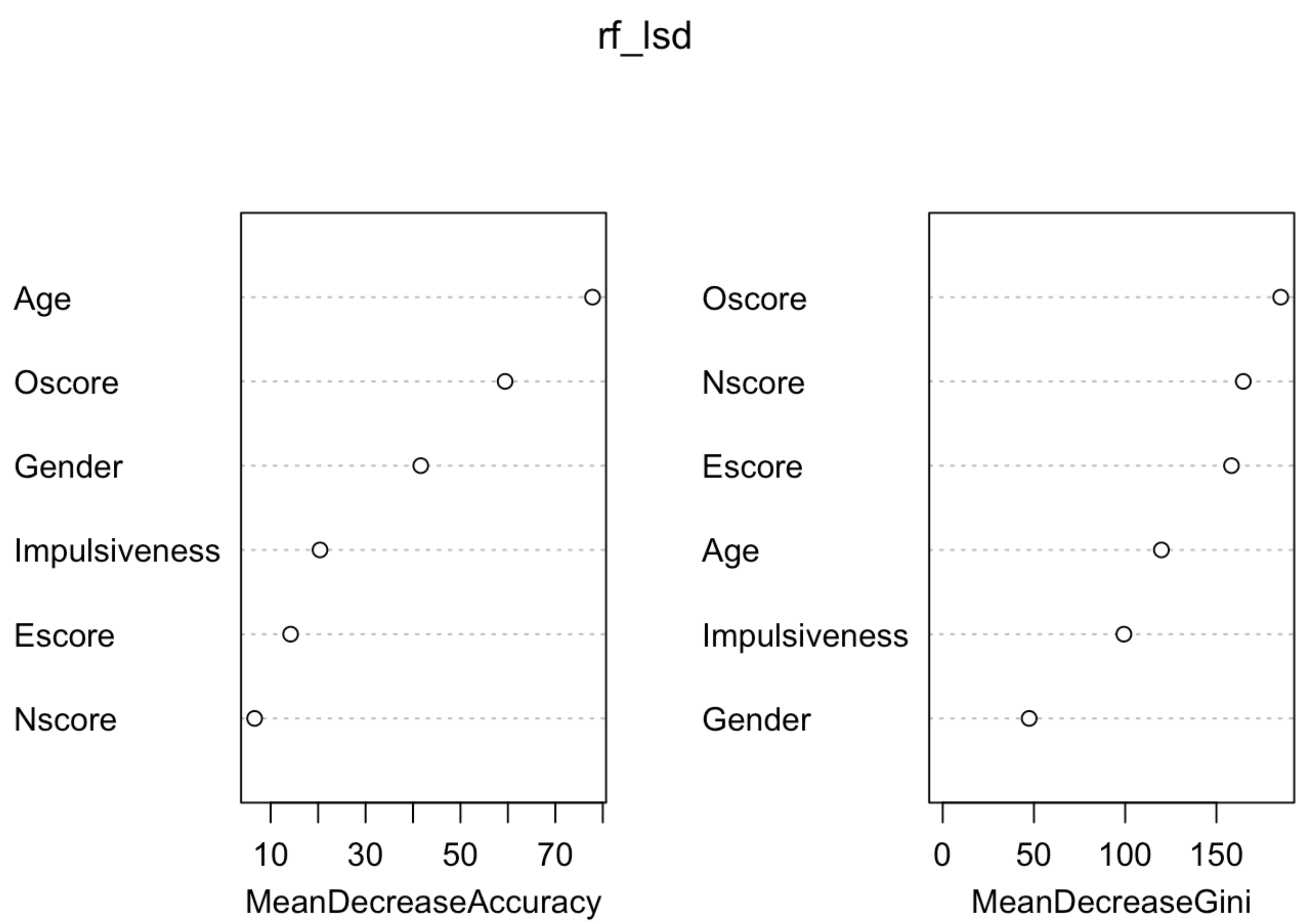

```
set.seed(365)

rf_lsd <- randomForest(LSD~.,
                    data = lsd,
                    importance = TRUE,
                    mtry = 3,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_lsd)
```

```
##
## Call:
##  randomForest(formula = LSD ~ ., data = lsd, importance = TRUE,      mtry = 3, ntr
ee = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 23.66%
## Confusion matrix:
##      0   1 class.error
## 0 1143 185   0.1393072
## 1  261 296   0.4685817
```

```
f1(rf_lsd) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6154, Recall: 0.5314, F1 Score: 0.5703"
```

```
varImpPlot(rf_lsd)
```



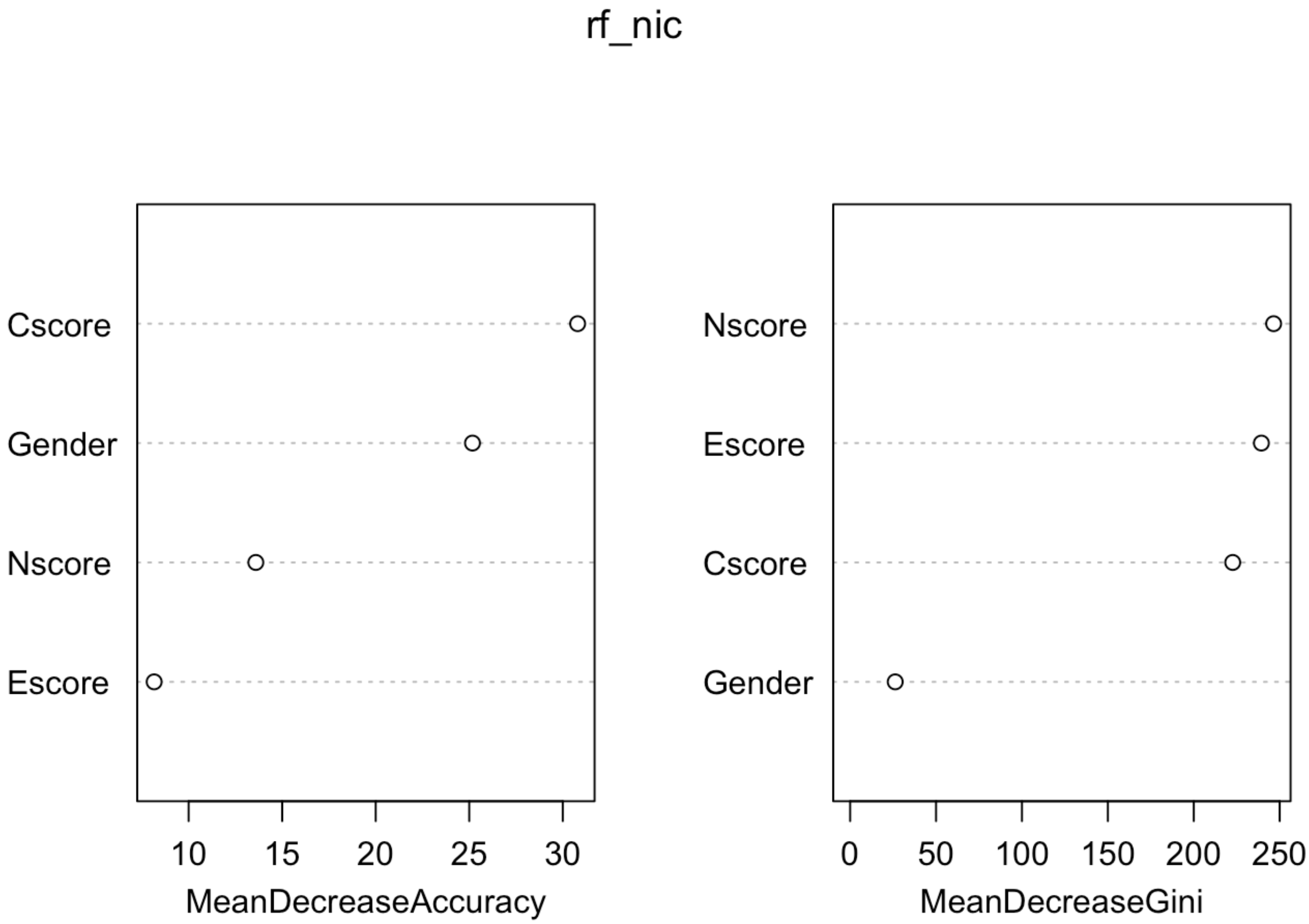
```r
set.seed(365)

rf_nic <- randomForest(Nicotine~.,
                    data = nicotine,
                    importance = TRUE,
                    mtry = 2,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_nic)
```

```
##
## Call:
##  randomForest(formula = Nicotine ~ ., data = nicotine, importance = TRUE,      mtr
y = 2, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 37.03%
## Confusion matrix:
##     0   1 class.error
## 0 189 432   0.6956522
## 1 266 998   0.2104430
```

```r
f1(rf_nic) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6979, Recall: 0.7896, F1 Score: 0.7409"
```

```r
varImpPlot(rf_nic)
```


rf_nic
Cscore
Gender
Nscore
Escore
10
15
20
25
30
MeanDecreaseAccuracy
Nscore
Escore
Cscore
Gender
0
50
100
150
200
250
MeanDecreaseGini