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("Cannabis", "Ecstasy")

#drugs = list("Cannabis", "Ecstasy")

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

data
```

```
## # A tibble: 1,885 × 32
##       ID     Age Gender Education Country Ethni…¹ Nscore   Escore  Oscore Ascore
##    <dbl>   <dbl>  <dbl>     <dbl>   <dbl>   <dbl>  <dbl>    <dbl>   <dbl>  <dbl>
##  1     1  0.498   0.482   -0.0592   0.961   0.126  0.313 -0.575   -0.583  -0.917
##  2     2 -0.0785 -0.482    1.98     0.961  -0.317 -0.678  1.94     1.44    0.761
##  3     3  0.498  -0.482   -0.0592   0.961  -0.317 -0.467  0.805   -0.847  -1.62
##  4     4 -0.952   0.482    1.16     0.961  -0.317 -0.149 -0.806   -0.0193  0.590
##  5     5  0.498   0.482    1.98     0.961  -0.317  0.735 -1.63    -0.452  -0.302
##  6     6  2.59    0.482   -1.23     0.249  -0.317 -0.678 -0.300   -1.56    2.04
##  7     7  1.09   -0.482    1.16    -0.570  -0.317 -0.467 -1.09    -0.452  -0.302
##  8     8  0.498  -0.482   -1.74     0.961  -0.317 -1.33   1.94    -0.847  -0.302
##  9     9  0.498   0.482   -0.0592   0.249  -0.317  0.630  2.57    -0.976   0.761
## 10    10  1.82   -0.482    1.16     0.961  -0.317 -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 <dbl>, Amyl <dbl>, Benzos <dbl>,
## #   Caff <dbl>, Cannabis <fct>, Choc <dbl>, Coke <dbl>, Crack <dbl>,
## #   Ecstasy <fct>, Heroin <dbl>, Ketamine <dbl>, Legalh <dbl>, LSD <dbl>,
## #   Meth <dbl>, Mushrooms <dbl>, Nicotine <dbl>, VSA <dbl>, Label <dbl>, and
## #   abbreviated variable name ¹Ethnicity
```

```r
# Feature Ranks for Cannabis -> 1:Age, 2:SS, 3:Oscore, 4:Cscore, 5:Nscore
# Feature Ranks for Ecstasy -> 1:SS, 2:Age, 3:Oscore, 4:Cscore, 5:Nscore

cannabis_top1 = data[c("Age", "Cannabis")]
ecstasy_top1 = data[c("SS", "Ecstasy")]

cannabis_top2 = data[c("Age", "SS", "Cannabis")]
ecstasy_top2 = data[c("Age", "SS", "Ecstasy")]

cannabis_top3 = data[c("Age", "Oscore", "SS", "Cannabis")]
ecstasy_top3 = data[c("Age", "Oscore", "SS", "Ecstasy")]

cannabis_top4 = data[c("Age", "Oscore", "Cscore", "SS", "Cannabis")]
ecstasy_top4 = data[c("Age", "Oscore", "Cscore", "SS", "Ecstasy")]

cannabis_top5 = data[c("Age", "Nscore", "Oscore", "Cscore", "SS", "Cannabis")]
ecstasy_top5 = data[c("Age", "Nscore", "Oscore", "Cscore", "SS", "Ecstasy")]
```

```r
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])

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

sprintf('Precision: %.4f, Recall: %.4f, F1 Score: %.4f', p, r,score)
}
```

2. Top 1 Features

```r
library(randomForest)
```

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

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

```
set.seed(365)

rf_can1 <- randomForest(Cannabis~.,
                    data = cannabis_top1,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_can1)
```

```
##
## Call:
##  randomForest(formula = Cannabis ~ ., data = cannabis_top1, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 27%
## Confusion matrix:
##     0    1 class.error
## 0 258  362   0.5838710
## 1 147 1118   0.1162055
```

```
f1(rf_can1) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.7554, Recall: 0.8838, F1 Score: 0.8146"
```

```
#varImpPlot(rf_can1) #ranking invalid
```

```
set.seed(365)

rf_xt1 <- randomForest(Ecstasy~.,
                    data = ecstasy_top1,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_xt1)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy_top1, importance = TRUE, 
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 31.3%
## Confusion matrix:
##     0   1 class.error
## 0 828 306   0.2698413
## 1 284 467   0.3781625
```

```
f1(rf_xt1) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6041, Recall: 0.6218, F1 Score: 0.6129"
```

```
#varImpPlot(rf_xt1) #ranking invalid
```

3. Top 2 Features

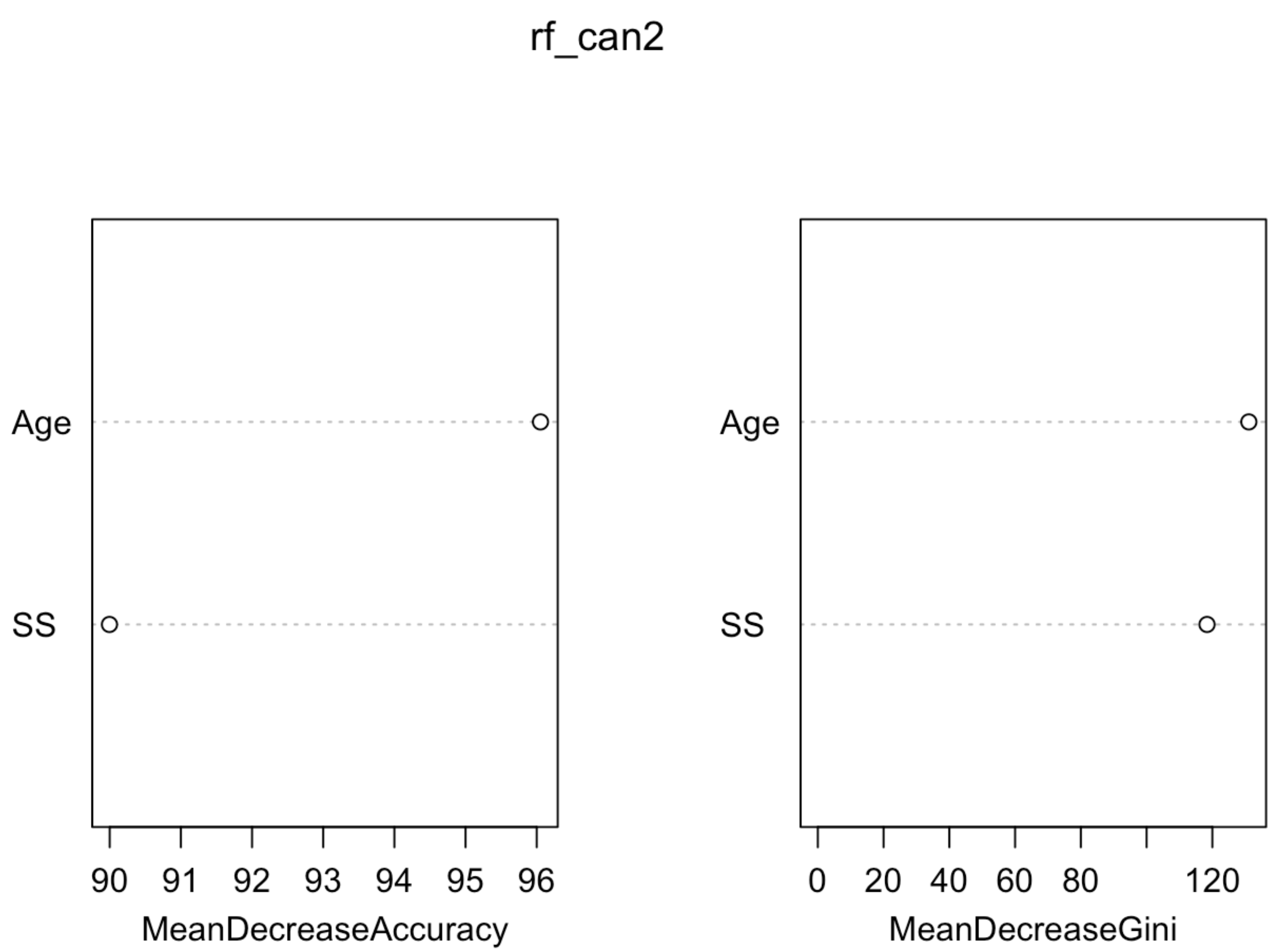
```
set.seed(365)

rf_can2 <- randomForest(Cannabis~.,
                    data = cannabis_top2,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_can2)
```

```
##
## Call:
##  randomForest(formula = Cannabis ~ ., data = cannabis_top2, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 22.81%
## Confusion matrix:
##     0    1 class.error
## 0 321  299   0.4822581
## 1 131 1134   0.1035573
```

```
f1(rf_can2) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.7913, Recall: 0.8964, F1 Score: 0.8406"
```

```
varImpPlot(rf_can2)
```



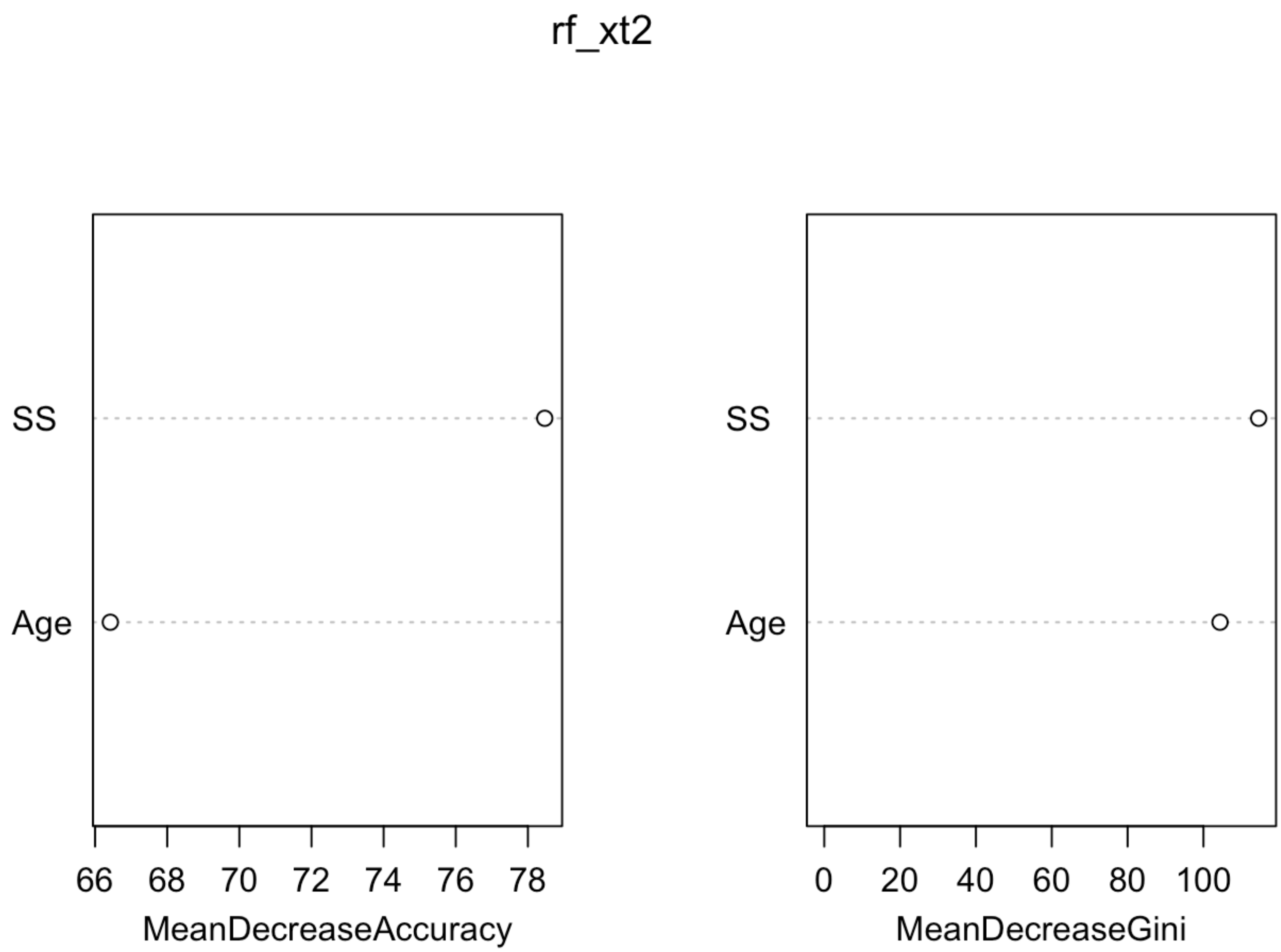
```
set.seed(365)

rf_xt2 <- randomForest(Ecstasy~.,
                    data = ecstasy_top2,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_xt2)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy_top2, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 29.6%
## Confusion matrix:
##     0   1 class.error
## 0 840 294   0.2592593
## 1 264 487   0.3515313
```

```
f1(rf_xt2) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6236, Recall: 0.6485, F1 Score: 0.6358"
```

```
varImpPlot(rf_xt2)
```



4. Top 3 Features

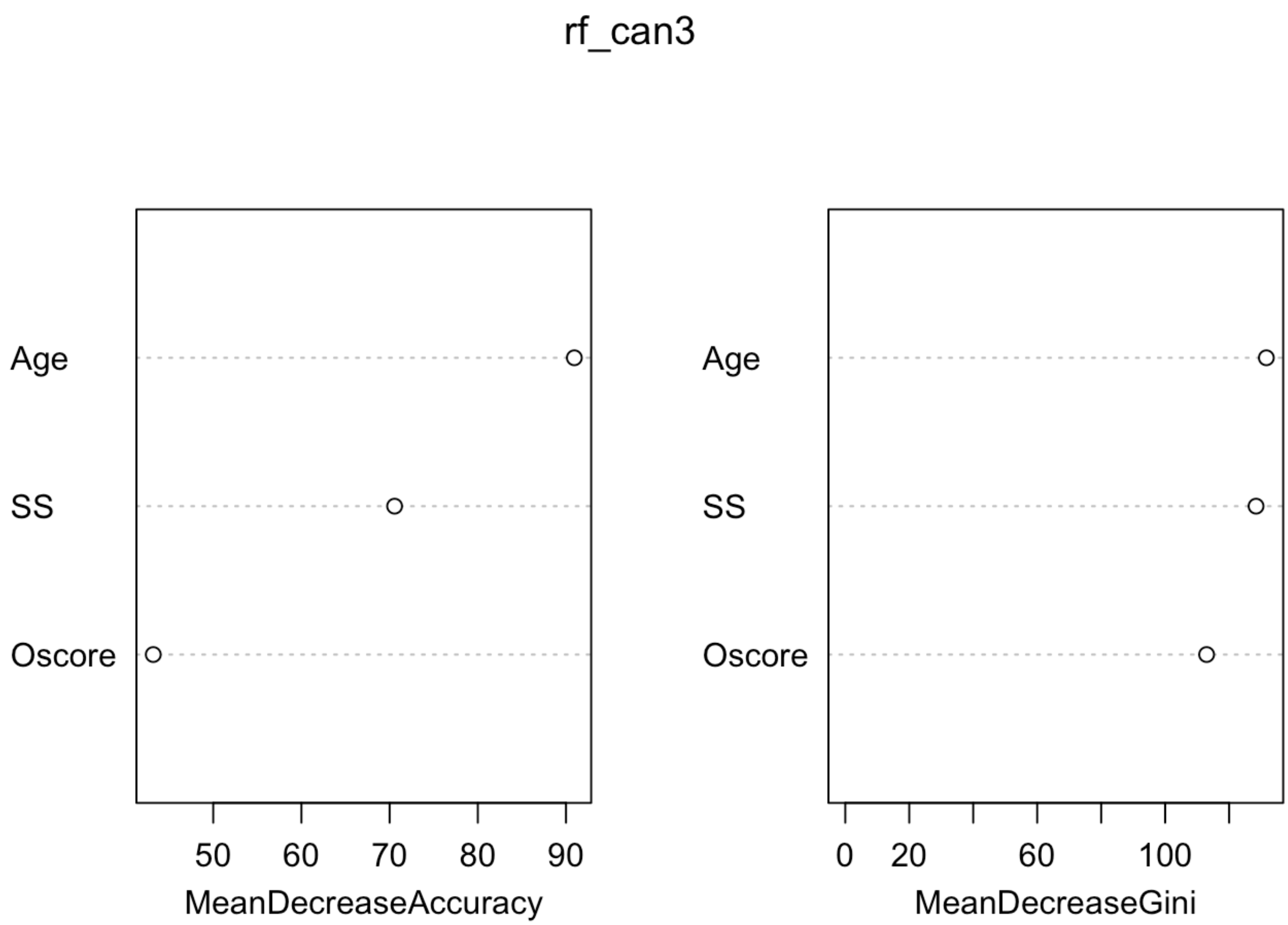
```
set.seed(365)

rf_can3 <- randomForest(Cannabis~.,
                    data = cannabis_top3,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_can3)
```

```
##
## Call:
##  randomForest(formula = Cannabis ~ ., data = cannabis_top3, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 23.24%
## Confusion matrix:
##     0    1 class.error
## 0 349  271   0.4370968
## 1 167 1098   0.1320158
```

```
f1(rf_can3) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.8020, Recall: 0.8680, F1 Score: 0.8337"
```

```
varImpPlot(rf_can3)
```



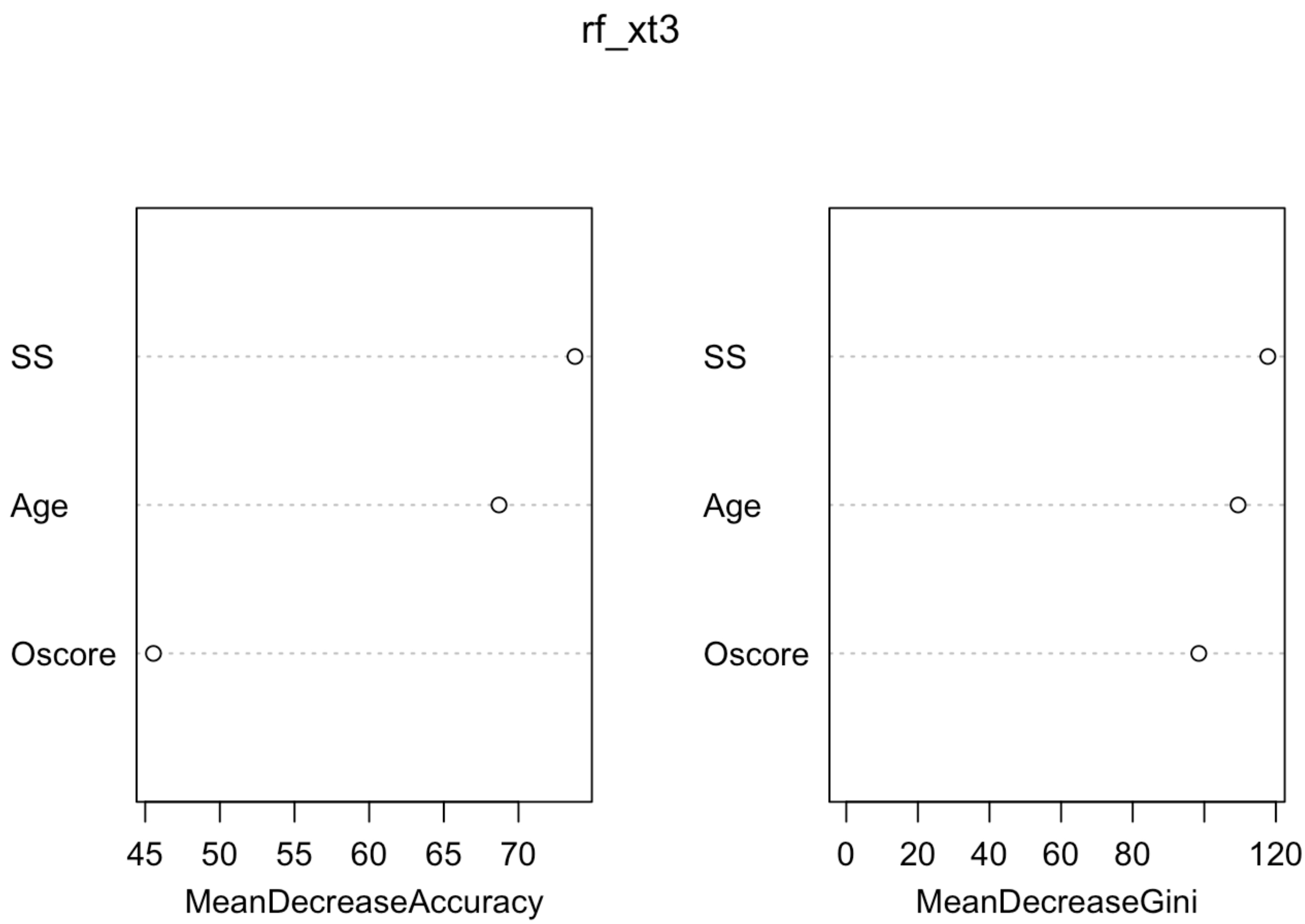
```
set.seed(365)

rf_xt3 <- randomForest(Ecstasy~.,
                    data = ecstasy_top3,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_xt3)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy_top3, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 27.59%
## Confusion matrix:
##     0   1 class.error
## 0 864 270   0.2380952
## 1 250 501   0.3328895
```

```
f1(rf_xt3) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6498, Recall: 0.6671, F1 Score: 0.6583"
```

```
varImpPlot(rf_xt3)
```



5. Top 4 Features

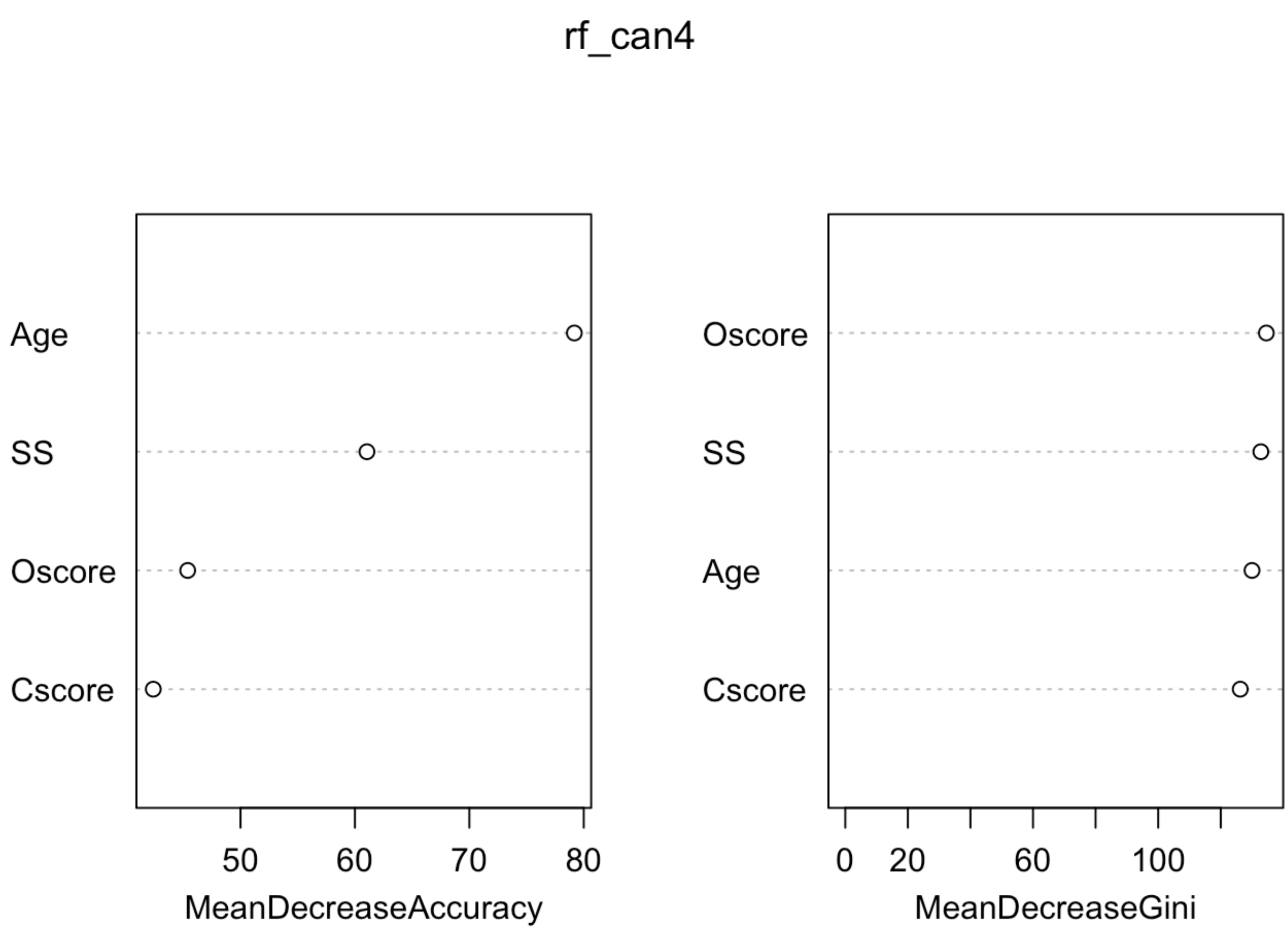
```
set.seed(365)

rf_can4 <- randomForest(Cannabis~.,
                    data = cannabis_top4,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_can4)
```

```
##
## Call:
##  randomForest(formula = Cannabis ~ ., data = cannabis_top4, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 22.44%
## Confusion matrix:
##     0    1 class.error
## 0 371  249   0.4016129
## 1 174 1091   0.1375494
```

```
f1(rf_can4) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.8142, Recall: 0.8625, F1 Score: 0.8376"
```

```
varImpPlot(rf_can4)
```



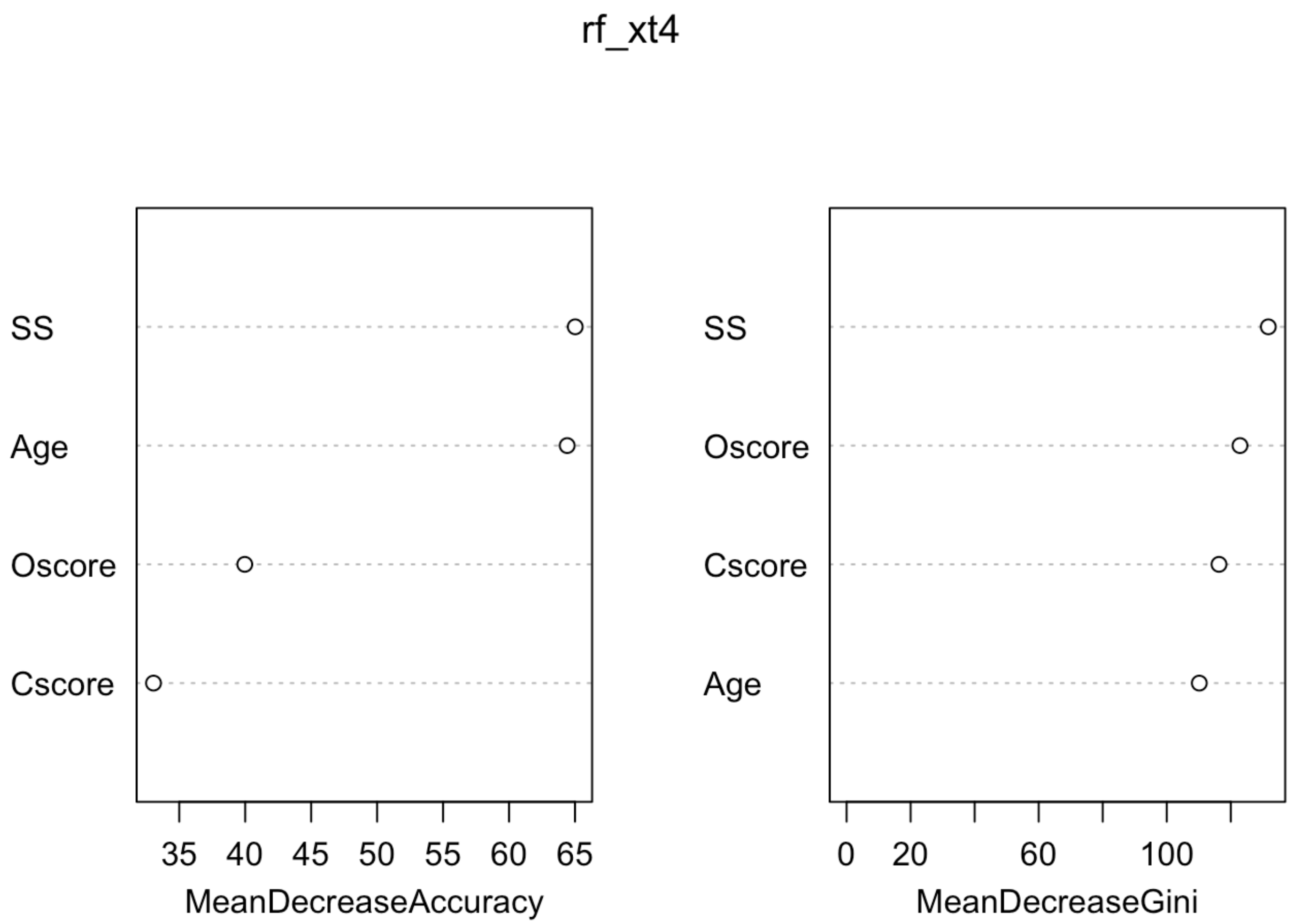
```
set.seed(365)

rf_xt4 <- randomForest(Ecstasy~.,
                    data = ecstasy_top4,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_xt4)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy_top4, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 27.37%
## Confusion matrix:
##     0   1 class.error
## 0 868 266   0.2345679
## 1 250 501   0.3328895
```

```
f1(rf_xt4) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6532, Recall: 0.6671, F1 Score: 0.6601"
```

```
varImpPlot(rf_xt4)
```



6. Top 5 Features

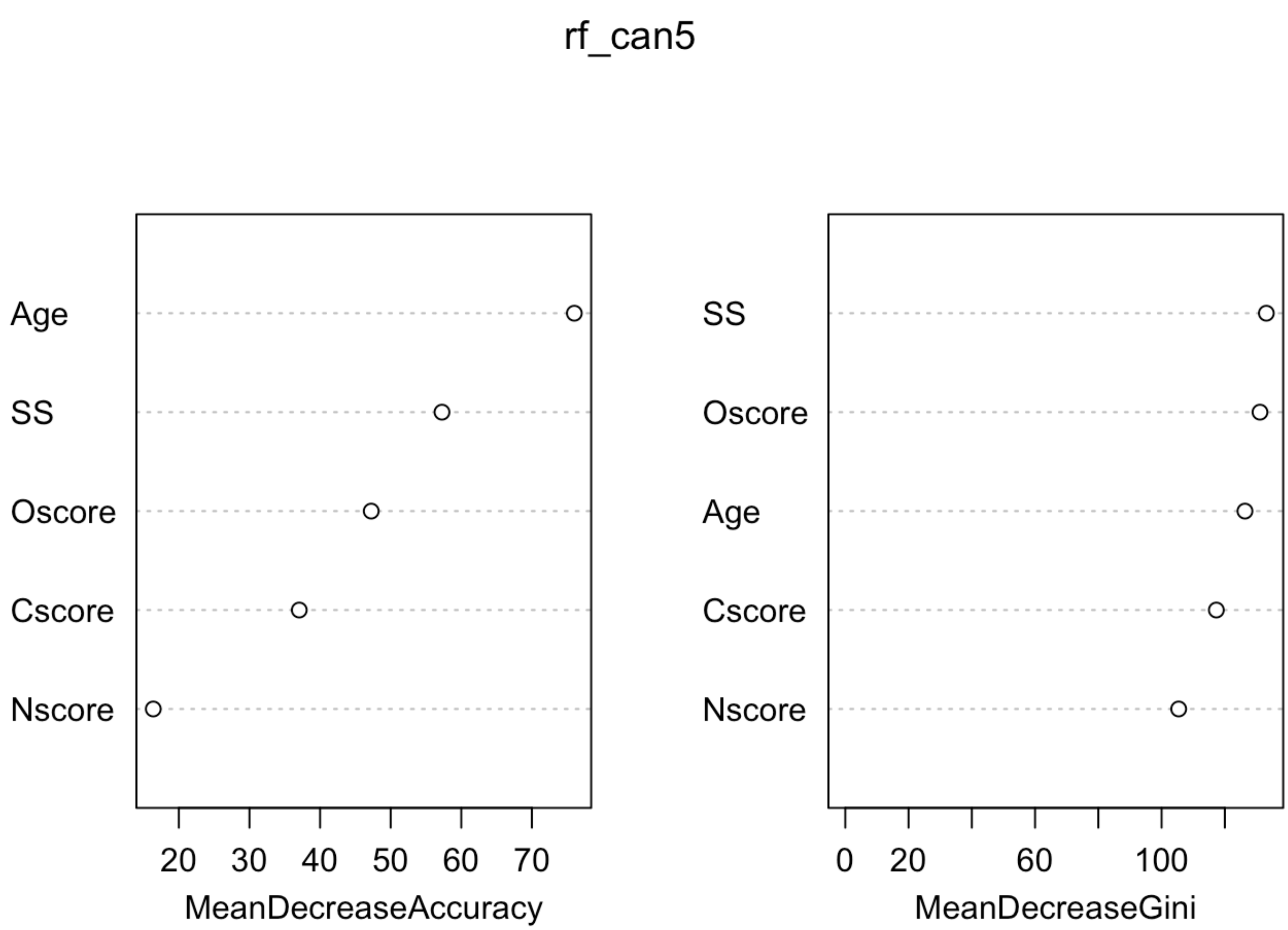
```
set.seed(365)

rf_can5 <- randomForest(Cannabis~.,
                    data = cannabis_top5,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_can5)
```

```
##
## Call:
##  randomForest(formula = Cannabis ~ ., data = cannabis_top5, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 21.06%
## Confusion matrix:
##     0    1 class.error
## 0 380  240   0.3870968
## 1 157 1108   0.1241107
```

```
f1(rf_can5) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.8220, Recall: 0.8759, F1 Score: 0.8481"
```

```
varImpPlot(rf_can5)
```



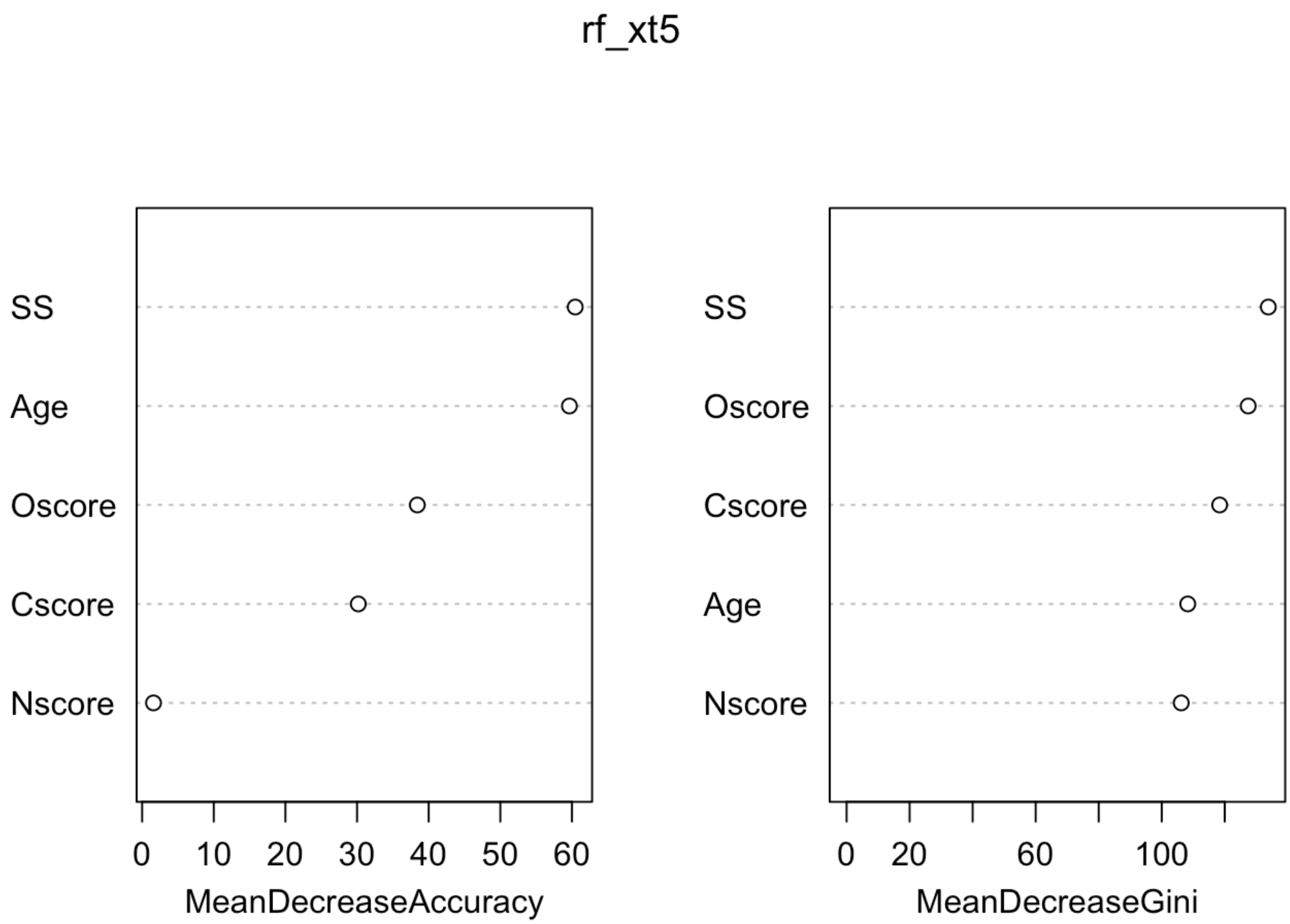
```
set.seed(365)

rf_xt5 <- randomForest(Ecstasy~.,
                    data = ecstasy_top5,
                    importance = TRUE,
                    mtry = 1,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_xt5)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy_top5, importance = TRUE,
mtry = 1, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 28.12%
## Confusion matrix:
##     0   1 class.error
## 0 871 263   0.2319224
## 1 267 484   0.3555260
```

```
f1(rf_xt5) #Positive Class F1 Score function
```

```
## [1] "Precision: 0.6479, Recall: 0.6445, F1 Score: 0.6462"
```

```
varImpPlot(rf_xt5)
```


rf_xt5
SS
Age
Oscore
Cscore
Nscore
0 10 20 30 40 50 60
MeanDecreaseAccuracy
SS
Oscore
Cscore
Age
Nscore
0 20 60 100
MeanDecreaseGini