1. Importing the data and helper function(s)

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

f_cols = c("Gender", "Country", "Ethnicity", "Alcohol", "Amphet", "Benzos", "Cannabis", "Ecstasy", "Legalh", "Nicotine")

#drugs = list("Alcohol", "Amphet", "Benzos", "Cannabis", "Ecstasy", "Legalh", "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 <fct>, Amphet <fct>, Amyl <dbl>, Benzos <fct>,
## #   Caff <dbl>, Cannabis <fct>, Choc <dbl>, Coke <dbl>, Crack <dbl>,
## #   Ecstasy <fct>, Heroin <dbl>, Ketamine <dbl>, Legalh <fct>, LSD <dbl>,
## #   Meth <dbl>, Mushrooms <dbl>, Nicotine <fct>, VSA <dbl>, Label <dbl>, and
## #   abbreviated variable names ¹Education, ²Ethnicity
```

```
alcohol = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Esc
ore", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "Alcohol")]

amphet = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Esco
re", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "Amphet")]

benzos = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Esco
re", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "Benzos")]

cannabis = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Es
core", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "Cannabis")]

ecstasy = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Esc
ore", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "Ecstasy")]

legalh = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Esco
re", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "Legalh")]

nicotine = data[c("Age", "Gender", "Education", "Country", "Ethnicity", "Nscore", "Es
core", "Oscore", "Ascore", "Cscore", "Impulsiveness", "SS", "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('Precision: %.4f', p)
#sprintf('Recall: %.4f', r)
#sprintf('Sensitivity: %.4f', se)
#sprintf('Specificity: %.4f', sp)
sprintf('F1 Score: %.4f', score)
}
```

2. Running a Random Forest

```
library(randomForest)
```

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

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

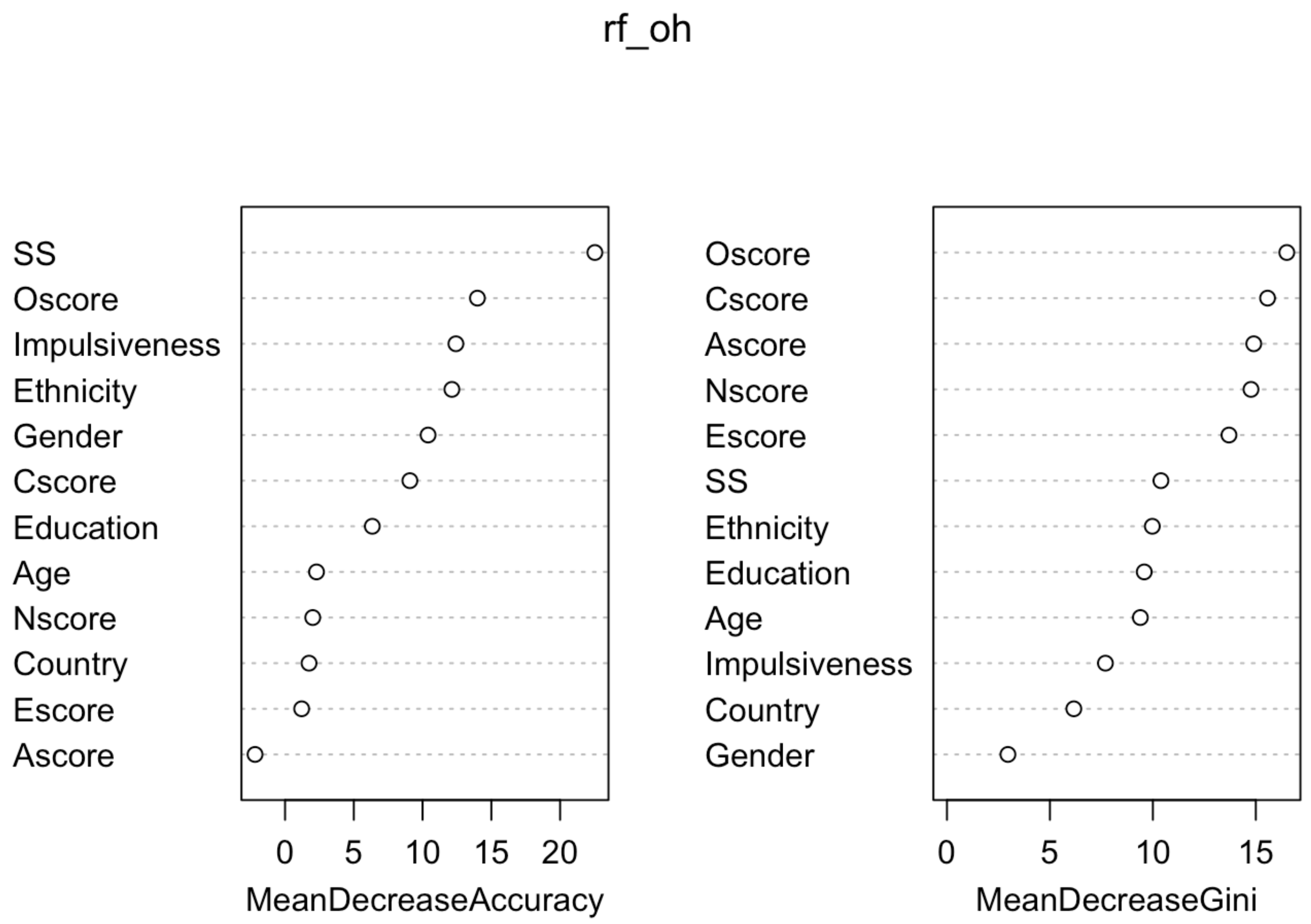
```
set.seed(365)

rf_oh <- randomForest(Alcohol~.,
                    data = alcohol,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_oh)
```

```
##
## Call:
##  randomForest(formula = Alcohol ~ ., data = alcohol, importance = TRUE,      mtry
= 4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 3.93%
## Confusion matrix:
##   0    1 class.error
## 0 0   68 1.000000000
## 1 6 1811 0.003302146
```

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

```
## [1] "F1 Score: 0.9800"
```

```
varImpPlot(rf_oh)
```



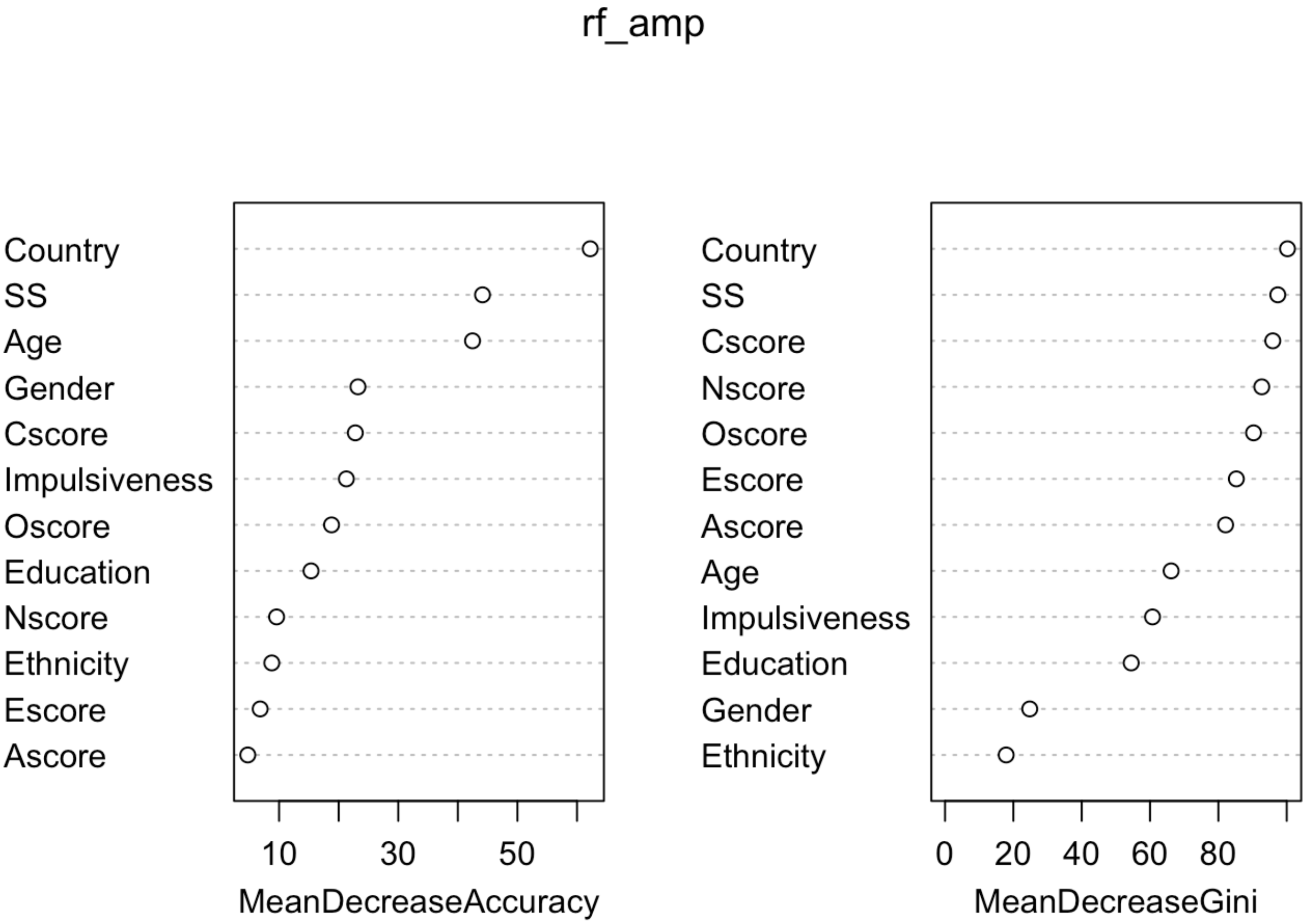
```
set.seed(365)

rf_amp <- randomForest(Amphet~.,
                    data = amphet,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_amp)
```

```
##
## Call:
##  randomForest(formula = Amphet ~ ., data = amphet, importance = TRUE,      mtry =
4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 27.96%
## Confusion matrix:
##     0   1 class.error
## 0 973 233   0.1932007
## 1 294 385   0.4329897
```

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

```
## [1] "F1 Score: 0.5937"
```

```
varImpPlot(rf_amp)
```



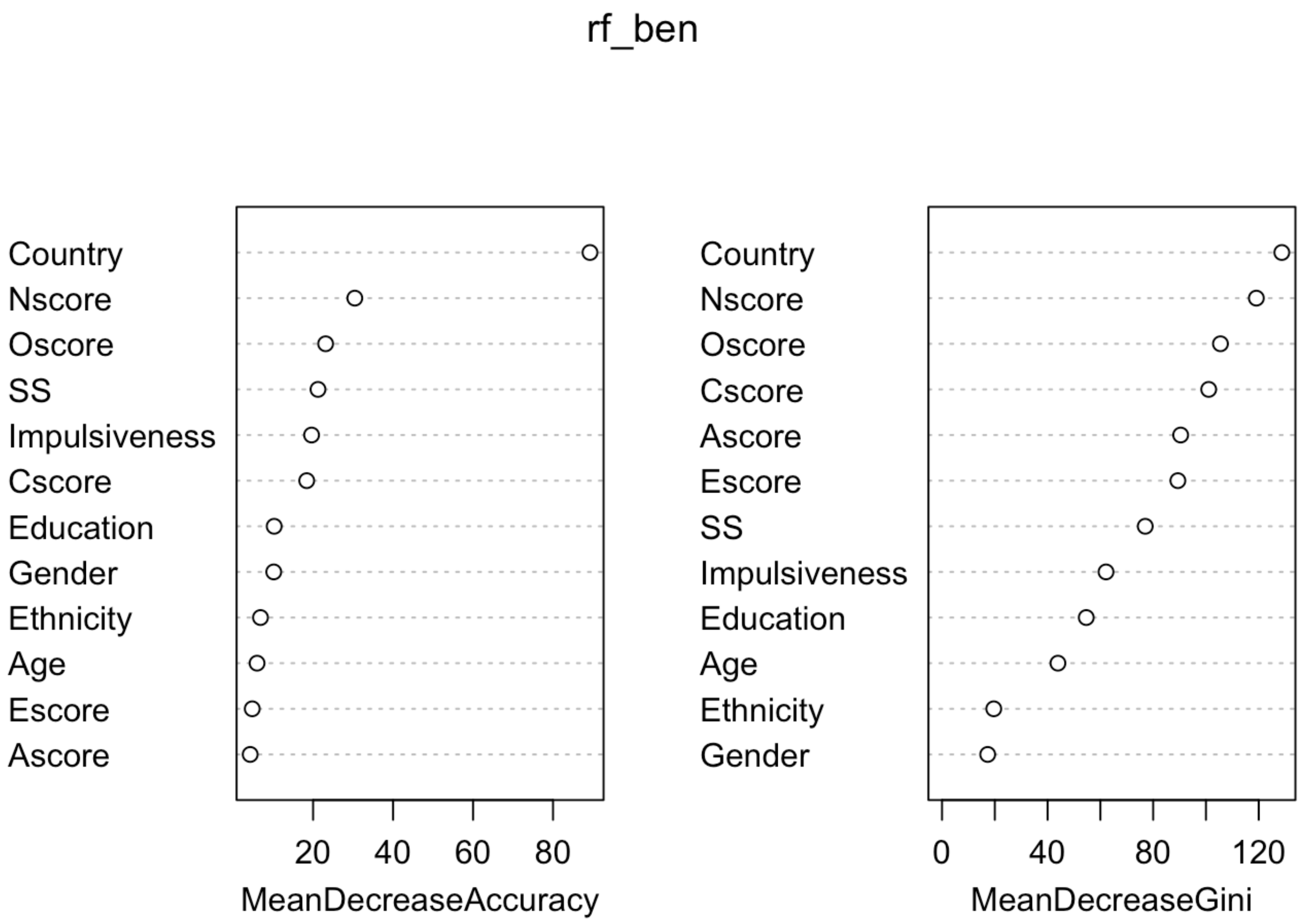
```
set.seed(365)

rf_ben <- randomForest(Benzos~.,
                    data = benzos,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_ben)
```

```
##
## Call:
##  randomForest(formula = Benzos ~ ., data = benzos, importance = TRUE,      mtry =
4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 30.13%
## Confusion matrix:
##     0   1 class.error
## 0 877 239   0.2141577
## 1 329 440   0.4278283
```

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

```
## [1] "F1 Score: 0.6077"
```

```
varImpPlot(rf_ben)
```



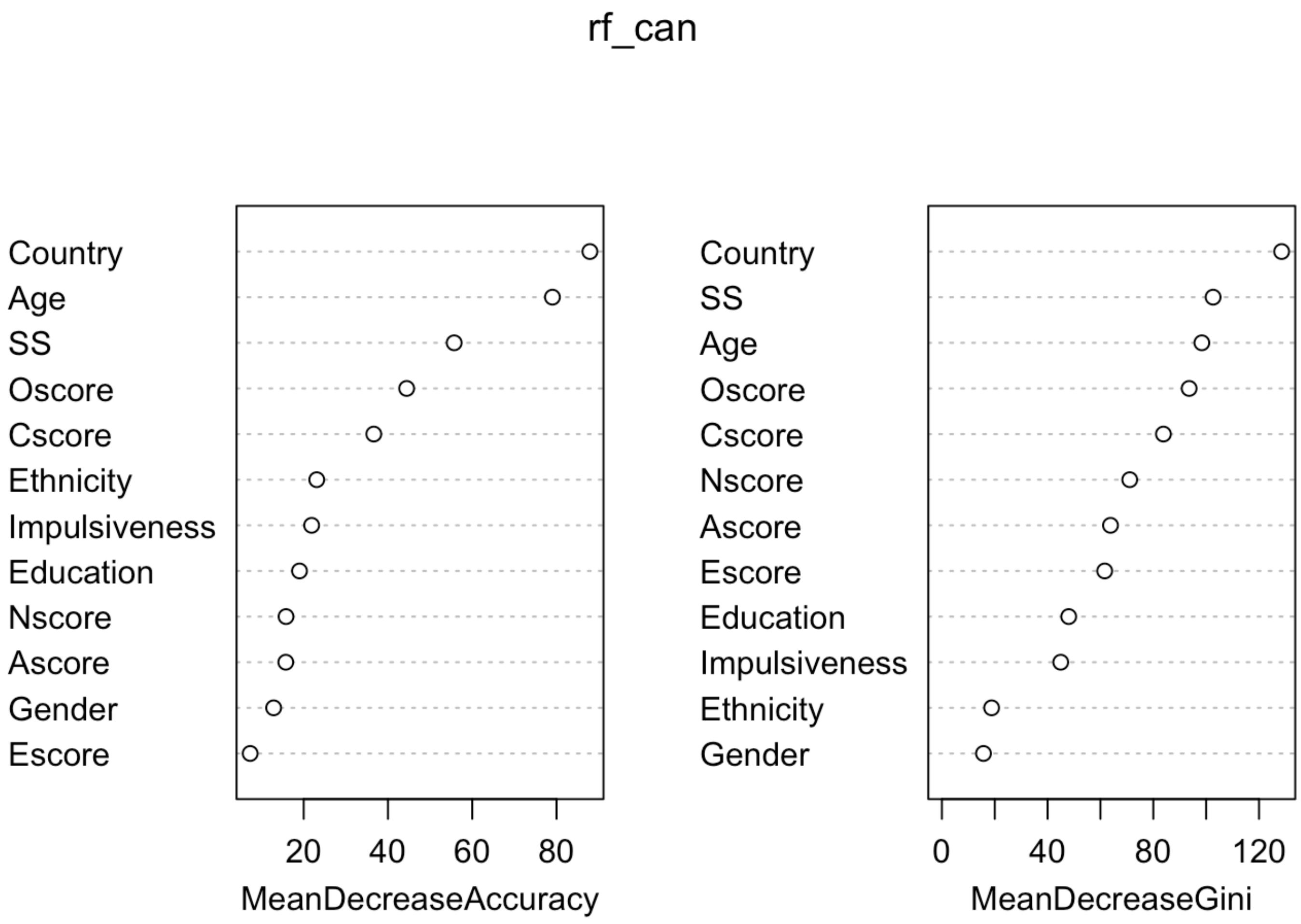
```
set.seed(365)

rf_can <- randomForest(Cannabis~.,
                    data = cannabis,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_can)
```

```
##
## Call:
##  randomForest(formula = Cannabis ~ ., data = cannabis, importance = TRUE,      mtr
y = 4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 18.99%
## Confusion matrix:
##     0    1 class.error
## 0 409  211   0.3403226
## 1 147 1118   0.1162055
```

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

```
## [1] "F1 Score: 0.8620"
```

```
varImpPlot(rf_can)
```



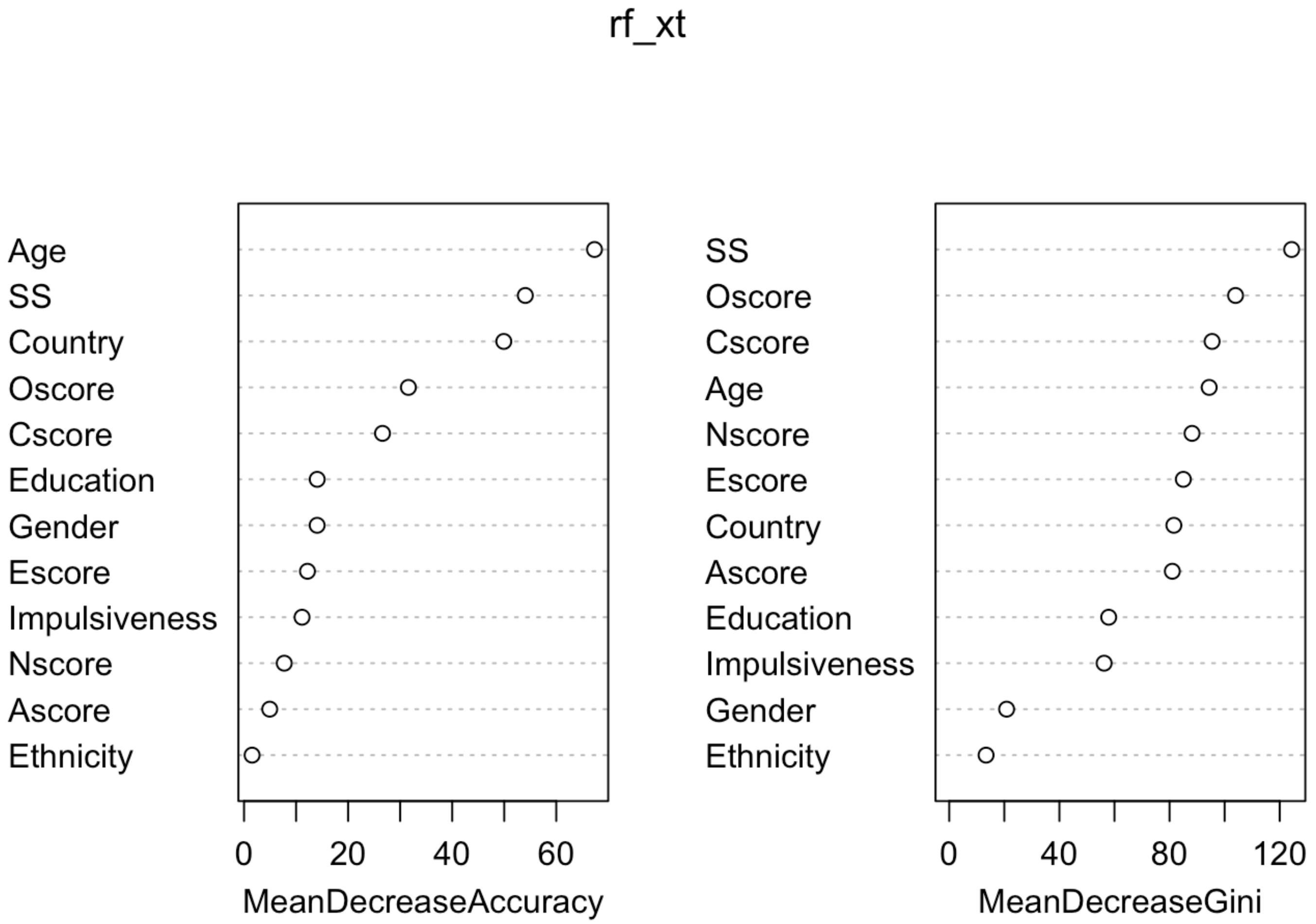
```
set.seed(365)

rf_xt <- randomForest(Ecstasy~.,
                    data = ecstasy,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_xt)
```

```
##
## Call:
##  randomForest(formula = Ecstasy ~ ., data = ecstasy, importance = TRUE,      mtry
= 4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 26.37%
## Confusion matrix:
##     0   1 class.error
## 0 882 252   0.2222222
## 1 245 506   0.3262317
```

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

```
## [1] "F1 Score: 0.6706"
```

```
varImpPlot(rf_xt)
```



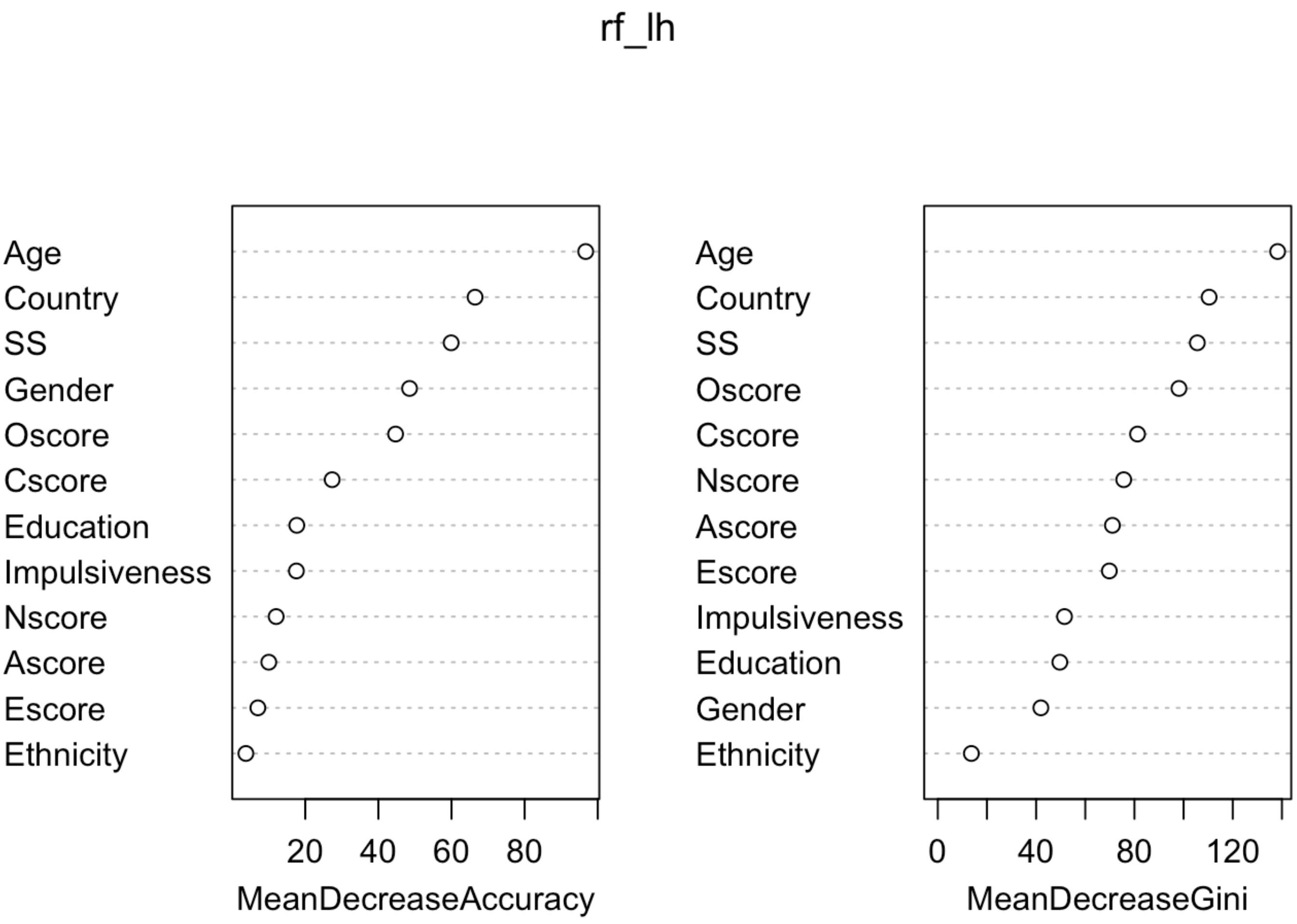
```
set.seed(365)

rf_lh <- randomForest(Legalh~.,
                    data = legalh,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_lh)
```

```
##
## Call:
##  randomForest(formula = Legalh ~ ., data = legalh, importance = TRUE,      mtry =
4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 21.06%
## Confusion matrix:
##     0   1 class.error
## 0 941 182   0.1620659
## 1 215 547   0.2821522
```

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

```
## [1] "F1 Score: 0.7337"
```

```
varImpPlot(rf_lh)
```



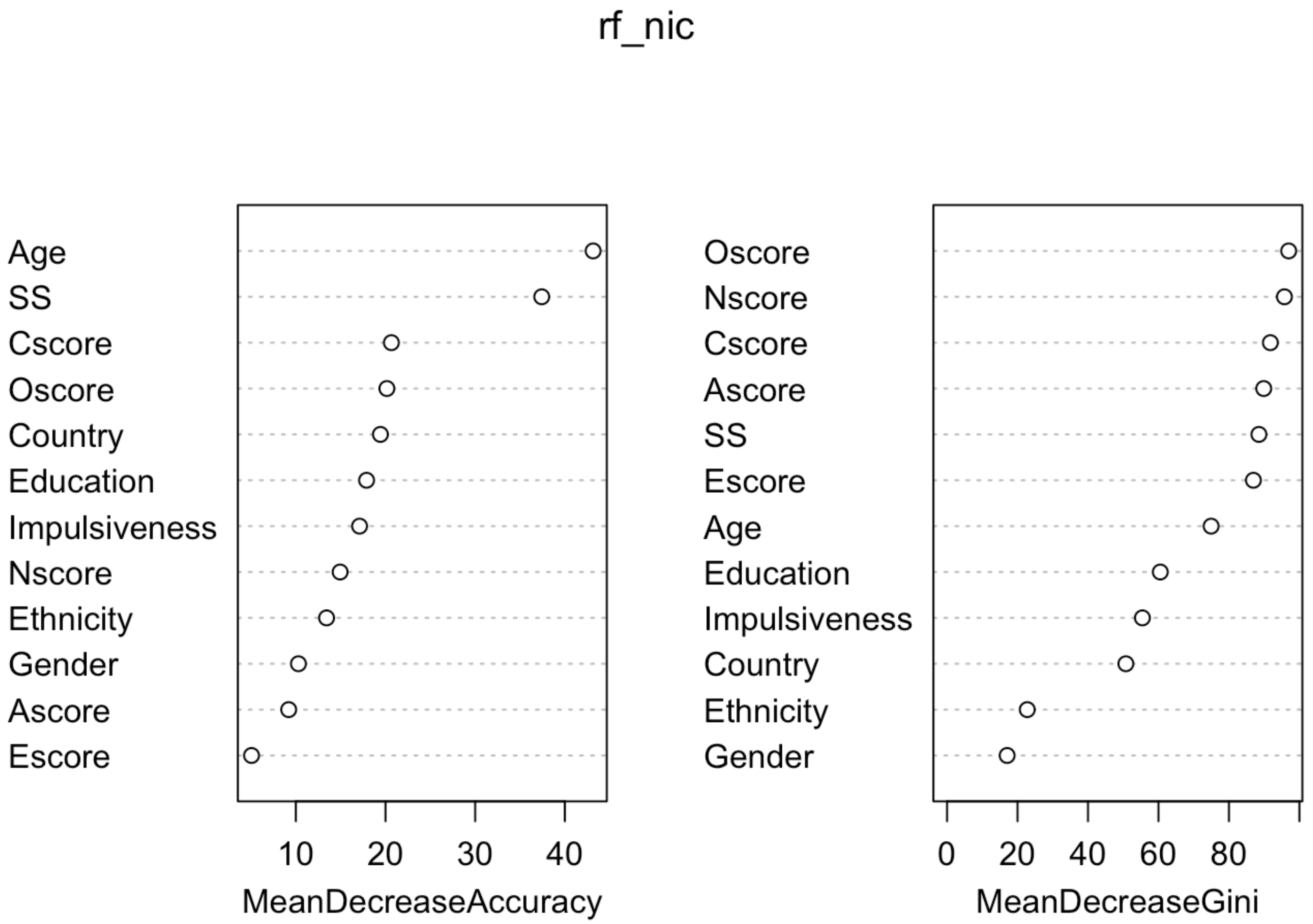
```
set.seed(365)

rf_nic <- randomForest(Nicotine~.,
                    data = nicotine,
                    importance = TRUE,
                    mtry = 4,
                    ntree = 1000,
                    CUTOFF = .6,
                    verbose = TRUE)
print(rf_nic)
```

```
##
## Call:
##  randomForest(formula = Nicotine ~ ., data = nicotine, importance = TRUE,      mtr
y = 4, ntree = 1000, CUTOFF = 0.6, verbose = TRUE)
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 27.06%
## Confusion matrix:
##     0    1 class.error
## 0 281  340   0.5475040
## 1 170 1094   0.1344937
```

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

```
## [1] "F1 Score: 0.8110"
```

```
varImpPlot(rf_nic)
```


rf_nic
Age
SS
Cscore
Oscore
Country
Education
Impulsiveness
Nscore
Ethnicity
Gender
Ascore
Escore
10
20
30
40
MeanDecreaseAccuracy
Oscore
Nscore
Cscore
Ascore
SS
Escore
Age
Education
Impulsiveness
Country
Ethnicity
Gender
0
20
40
60
80
MeanDecreaseGini