Assignment 4: Namit Shrivastava

Setup

```
library(caret)
library(randomForest)
library(partykit)
library(pdp)
library(iml)
```

Data

Here we use data from the UCI Machine Learning repository on drug consumption. The data contains records for 1885 respondents with personality measurements (e.g. Big-5), level of education, age, gender, country of residence and ethnicity as features. In addition, information on the usage of 18 drugs is included.

Source: https://archive.ics.uci.edu/ml/datasets/Drug+consumption+%28quantified%29

```
library(mlforsocialscience)
data(drugs)
```

---

1) Predicting drug usage

a) Prepare an outcome variable. For this you can choose from the variables on drug consumption and pick one drug (or a combination of drugs) as the prediction objective. The resulting variable should be of class `factor`, but it can have more than two categories if needed.

```
str(drugs)
```

```
'data.frame':   1885 obs. of  32 variables:
 $ ID                : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Age               : Factor w/ 6 levels "-0.95197","-0.07854",..: 3 2 3 1 3 6 4 3 3 5 ...
 $ Gender            : Factor w/ 2 levels "-0.48246","0.48246": 2 1 1 2 2 2 1 1 2 1 ...
 $ Education         : Factor w/ 9 levels "-2.43591","-1.7379",..: 6 9 6 8 9 4 8 2 6 8 ...
 $ Country           : Factor w/ 7 levels "-0.57009","-0.46841",..: 7 7 7 7 7 6 1 7 6 7 ...
 $ Ethnicity         : Factor w/ 7 levels "-1.10702","-0.50212",..: 6 3 3 3 3 3 3 3 3 3 ...
 $ Neuroticism       : num  0.313 -0.678 -0.467 -0.149 0.735 ...
 $ Extraversion      : num  -0.575 1.939 0.805 -0.806 -1.633 ...
 $ Openness          : num  -0.5833 1.4353 -0.8473 -0.0193 -0.4517 ...
 $ Agreeableness     : num  -0.917 0.761 -1.621 0.59 -0.302 ...
 $ Conscientiousness: num  -0.00665 -0.14277 -1.0145 0.58489 1.30612 ...
 $ Impulsive         : num  -0.217 -0.711 -1.38 -1.38 -0.217 ...
 $ SS                : num  -1.181 -0.216 0.401 -1.181 -0.216 ...
 $ Alcohol           : chr  "CL5" "CL5" "CL6" "CL4" ...
 $ Amphet            : chr  "CL2" "CL2" "CL0" "CL0" ...
 $ Amyl              : chr  "CL0" "CL2" "CL0" "CL0" ...
 $ Benzos            : chr  "CL2" "CL0" "CL0" "CL3" ...
 $ Caff              : chr  "CL6" "CL6" "CL6" "CL5" ...
 $ Cannabis          : chr  "CL0" "CL4" "CL3" "CL2" ...
 $ Choc              : chr  "CL5" "CL6" "CL4" "CL4" ...
 $ Coke              : chr  "CL0" "CL3" "CL0" "CL2" ...
 $ Crack             : chr  "CL0" "CL0" "CL0" "CL0" ...
 $ Ecstasy           : chr  "CL0" "CL4" "CL0" "CL0" ...
 $ Heroin            : chr  "CL0" "CL0" "CL0" "CL0" ...
 $ Ketamine          : chr  "CL0" "CL2" "CL0" "CL2" ...
 $ Legalh            : chr  "CL0" "CL0" "CL0" "CL0" ...
 $ LSD               : chr  "CL0" "CL2" "CL0" "CL0" ...
 $ Meth              : chr  "CL0" "CL3" "CL0" "CL0" ...
 $ Mushrooms         : chr  "CL0" "CL0" "CL1" "CL0" ...
 $ Nicotine          : chr  "CL2" "CL4" "CL0" "CL2" ...
 $ Semer             : chr  "CL0" "CL0" "CL0" "CL0" ...
 $ VSA               : chr  "CL0" "CL0" "CL0" "CL0" ...
```

```
# Examining the Cannabis variable which appears to use "CL" categories
table(drugs$Cannabis)
```

```
CL0 CL1 CL2 CL3 CL4 CL5 CL6
```

```
413 207 266 211 140 185 463
```

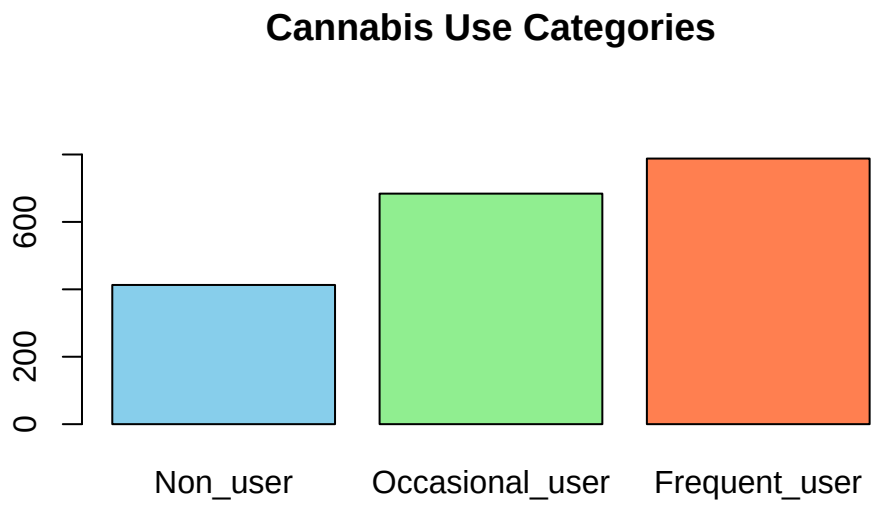
```r
# Creating a meaningful cannabis use outcome variable
drugs$CannabisUse <- factor(
  ifelse(drugs$Cannabis == "CL0", "Non_user",
         ifelse(drugs$Cannabis %in% c("CL1", "CL2", "CL3"),
                "Occasional_user",
                "Frequent_user")),
  levels = c("Non_user", "Occasional_user", "Frequent_user")
)

# Checking the updated factor levels
table(drugs$CannabisUse)
```

```
       Non_user Occasional_user   Frequent_user
            413             684             788
```

```r
prop.table(table(drugs$CannabisUse)) * 100
```

```
       Non_user Occasional_user   Frequent_user
       21.90981        36.28647        41.80371
```

```r
# Visualizing the distribution
barplot(table(drugs$CannabisUse),
        main="Cannabis Use Categories",
        col=c("skyblue", "lightgreen", "coral"),
        ylim=c(0, max(table(drugs$CannabisUse))*1.2))
```



b) Next split the data into a training and a test part.

```
# Set seed for reproducibility
set.seed(9574)

# So creating a stratified partitioning based on the CannabisUse variable
# which ensures balanced distribution of classes in both train and test sets
inTrain <- createDataPartition(drugs$CannabisUse,
                               p = .8,
                               list = FALSE,
                               times = 1)

# Creating training and test datasets
drugs_train <- drugs[inTrain,]
drugs_test <- drugs[-inTrain,]

# Verifying whether the split worked correctly
cat("Training set dimensions:", dim(drugs_train), "\n")
```

```
Training set dimensions: 1510 33
```

```
cat("Test set dimensions:", dim(drugs_test), "\n")
```

```
Test set dimensions: 375 33
```

```
# Checking class distribution in both sets to ensure stratification worked
print("Class distribution in training set:")
```

```
[1] "Class distribution in training set:"
```

```
prop.table(table(drugs_train$CannabisUse)) * 100
```

```
      Non_user Occasional_user   Frequent_user
      21.92053        36.29139        41.78808
```

```
print("Class distribution in test set:")
```

```
[1] "Class distribution in test set:"
```

```
prop.table(table(drugs_test$CannabisUse)) * 100
```

```
      Non_user Occasional_user   Frequent_user
      21.86667        36.26667        41.86667
```

c) Specify the evaluation method for the `train()` function of `caret` with 10-fold cross-validation.

```
ctrl <- trainControl(
  method = "cv", # 10-fold cross-validation
  number = 10, # Number of folds
  classProbs = TRUE, # Calculating class probabilities
  summaryFunction = multiClassSummary,  # For multi-class problems
  verboseIter = TRUE,  # Showing progress
  savePredictions = "final" # Saving predictions for ROC curves later
)
```

d) Specify a grid object for tuning a random forest model.

```
rf_grid <- expand.grid(
  mtry = c(2, 4, 6, 8) # Number of variables randomly sampled at each split
)
```

e) Use `train()` from `caret` in order to grow the forest. Do not use any of the other drugs as predictors in this model. Determine the best model based on the tuning results.

```
# First, I will exclude drug-related columns
drug_cols <- grep("^(Alcohol|Amphet|Amyl|Benzos|Caff|Cannabis
                  |Choc|Coke|Crack|Ecstasy|Heroin|Ketamine|Legalh|LSD|Meth
                  |Mushrooms|Nicotine|Semer|VSA)$",
                  names(drugs_train), value = TRUE)

# Creating a formula that includes all predictors except other drugs and the outcome
predictors <- setdiff(names(drugs_train), c(drug_cols, "CannabisUse", "ID"))
formula_str <- paste("CannabisUse ~", paste(predictors, collapse = " + "))
rf_formula <- as.formula(formula_str)

# Set seed for reproducibility
set.seed(123)

# Train the random forest model
rf_model <- train(
  rf_formula,
  data = drugs_train,
  method = "rf", # Random Forest
  trControl = ctrl, # Training control parameters
  tuneGrid = rf_grid, # Tuning grid
  importance = TRUE, # Calculate variable importance
  ntree = 500 # Number of trees to grow
)
```

```
+ Fold01: mtry=2
- Fold01: mtry=2
+ Fold01: mtry=4
- Fold01: mtry=4
+ Fold01: mtry=6
- Fold01: mtry=6
+ Fold01: mtry=8
- Fold01: mtry=8
+ Fold02: mtry=2
```

```
- Fold02: mtry=2
+ Fold02: mtry=4
- Fold02: mtry=4
+ Fold02: mtry=6
- Fold02: mtry=6
+ Fold02: mtry=8
- Fold02: mtry=8
+ Fold03: mtry=2
- Fold03: mtry=2
+ Fold03: mtry=4
- Fold03: mtry=4
+ Fold03: mtry=6
- Fold03: mtry=6
+ Fold03: mtry=8
- Fold03: mtry=8
+ Fold04: mtry=2
- Fold04: mtry=2
+ Fold04: mtry=4
- Fold04: mtry=4
+ Fold04: mtry=6
- Fold04: mtry=6
+ Fold04: mtry=8
- Fold04: mtry=8
+ Fold05: mtry=2
- Fold05: mtry=2
+ Fold05: mtry=4
- Fold05: mtry=4
+ Fold05: mtry=6
- Fold05: mtry=6
+ Fold05: mtry=8
- Fold05: mtry=8
+ Fold06: mtry=2
- Fold06: mtry=2
+ Fold06: mtry=4
- Fold06: mtry=4
+ Fold06: mtry=6
- Fold06: mtry=6
+ Fold06: mtry=8
- Fold06: mtry=8
+ Fold07: mtry=2
- Fold07: mtry=2
+ Fold07: mtry=4
- Fold07: mtry=4
```

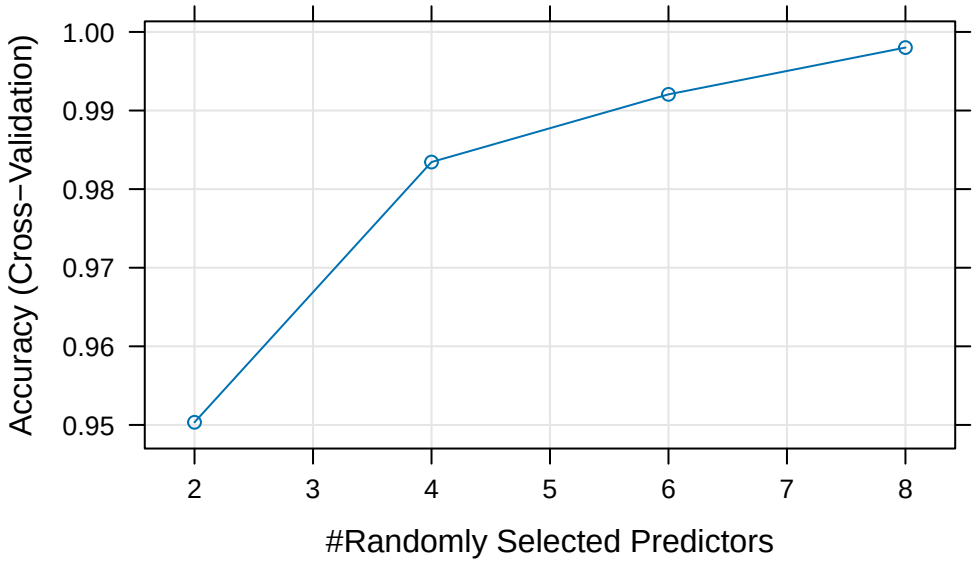
```
+ Fold07: mtry=6
- Fold07: mtry=6
+ Fold07: mtry=8
- Fold07: mtry=8
+ Fold08: mtry=2
- Fold08: mtry=2
+ Fold08: mtry=4
- Fold08: mtry=4
+ Fold08: mtry=6
- Fold08: mtry=6
+ Fold08: mtry=8
- Fold08: mtry=8
+ Fold09: mtry=2
- Fold09: mtry=2
+ Fold09: mtry=4
- Fold09: mtry=4
+ Fold09: mtry=6
- Fold09: mtry=6
+ Fold09: mtry=8
- Fold09: mtry=8
+ Fold10: mtry=2
- Fold10: mtry=2
+ Fold10: mtry=4
- Fold10: mtry=4
+ Fold10: mtry=6
- Fold10: mtry=6
+ Fold10: mtry=8
- Fold10: mtry=8
Aggregating results
Selecting tuning parameters
Fitting mtry = 8 on full training set
```

```
# tuning results
print(rf_model)
```

```
Random Forest

1510 samples
  14 predictor
   3 classes: 'Non_user', 'Occasional_user', 'Frequent_user'

No pre-processing
```

```
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1358, 1359, 1359, 1359, 1359, 1360, ...
Resampling results across tuning parameters:

  mtry  logLoss    AUC        prAUC      Accuracy   Kappa      Mean_F1
  2     0.4333048  0.9854448  0.9472504  0.9503350  0.9221657  0.9379289
  4     0.2467130  0.9987561  0.9739293  0.9834436  0.9741997  0.9799771
  6     0.1601589  0.9998766  0.9760669  0.9920573  0.9876287  0.9904852
  8     0.1091062  0.9999914  0.9720379  0.9980088  0.9969049  0.9976622
  Mean_Sensitivity  Mean_Specificity  Mean_Pos_Pred_Value  Mean_Neg_Pred_Value
  0.9278697         0.9733163         0.9563411            0.9791282
  0.9747772         0.9907461         0.9866669            0.9931187
  0.9878788         0.9955169         0.9937039            0.9966870
  0.9969697         0.9988593         0.9984535            0.9991620
  Mean_Precision  Mean_Recall  Mean_Detection_Rate  Mean_Balanced_Accuracy
  0.9563411       0.9278697    0.3167783            0.9505930
  0.9866669       0.9747772    0.3278145            0.9827617
  0.9937039       0.9878788    0.3306858            0.9916978
  0.9984535       0.9969697    0.3326696            0.9979145

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 8.
```

```r
plot(rf_model)
```




```
# Showing the best tuning parameter
cat("Best mtry value:", rf_model$bestTune$mtry, "\n")
```

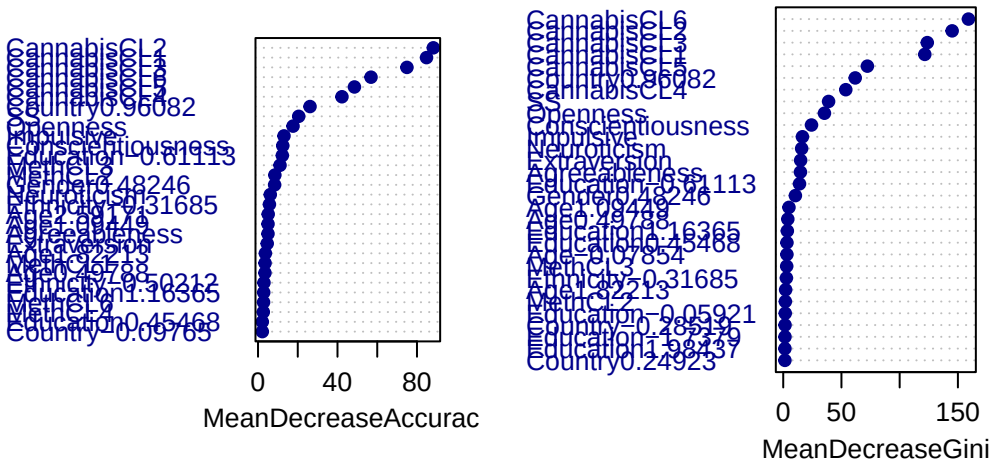
```
Best mtry value: 8
```

```
# Getting the final model performance
print(rf_model$results)
```

```
  mtry   logLoss       AUC     prAUC  Accuracy     Kappa   Mean_F1
1    2 0.4333048 0.9854448 0.9472504 0.9503350 0.9221657 0.9379289
2    4 0.2467130 0.9987561 0.9739293 0.9834436 0.9741997 0.9799771
3    6 0.1601589 0.9998766 0.9760669 0.9920573 0.9876287 0.9904852
4    8 0.1091062 0.9999914 0.9720379 0.9980088 0.9969049 0.9976622
  Mean_Sensitivity Mean_Specificity Mean_Pos_Pred_Value Mean_Neg_Pred_Value
1        0.9278697        0.9733163           0.9563411           0.9791282
2        0.9747772        0.9907461           0.9866669           0.9931187
3        0.9878788        0.9955169           0.9937039           0.9966870
4        0.9969697        0.9988593           0.9984535           0.9991620
  Mean_Precision Mean_Recall Mean_Detection_Rate Mean_Balanced_Accuracy
1      0.9563411   0.9278697           0.3167783              0.9505930
2      0.9866669   0.9747772           0.3278145              0.9827617
3      0.9937039   0.9878788           0.3306858              0.9916978
```

```
4      0.9984535   0.9969697           0.3326696              0.9979145
    logLossSD        AUCSD     prAUCSD  AccuracySD     KappaSD   Mean_F1SD
1 0.015284214 6.815225e-03 0.011972721 0.014367279 0.022777868 0.018632544
2 0.013123784 1.776739e-03 0.004691686 0.008966977 0.014044128 0.011282534
3 0.009897133 2.564277e-04 0.003111460 0.009254819 0.014439974 0.011205152
4 0.009289492 2.706966e-05 0.006950717 0.004474965 0.006958341 0.005266194
  Mean_SensitivitySD Mean_SpecificitySD Mean_Pos_Pred_ValueSD
1        0.020955786        0.007866096           0.012962855
2        0.013713674        0.004908988           0.007360011
3        0.014125372        0.005204909           0.007323431
4        0.006817662        0.002561673           0.003470588
  Mean_Neg_Pred_ValueSD Mean_PrecisionSD Mean_RecallSD Mean_Detection_RateSD
1           0.005986931      0.012962855   0.020955786           0.004789093
2           0.003638274      0.007360011   0.013713674           0.002988992
3           0.003829872      0.007323431   0.014125372           0.003084940
4           0.001880334      0.003470588   0.006817662           0.001491655
  Mean_Balanced_AccuracySD
1              0.014390058
2              0.009310418
3              0.009664719
4              0.004689656
```

```r
# Plotting variable importance for the final model
varImpPlot(rf_model$finalModel,
           main = "Variable Importance",
           pch = 19,
           col = "darkblue",
           cex = 0.8)
```

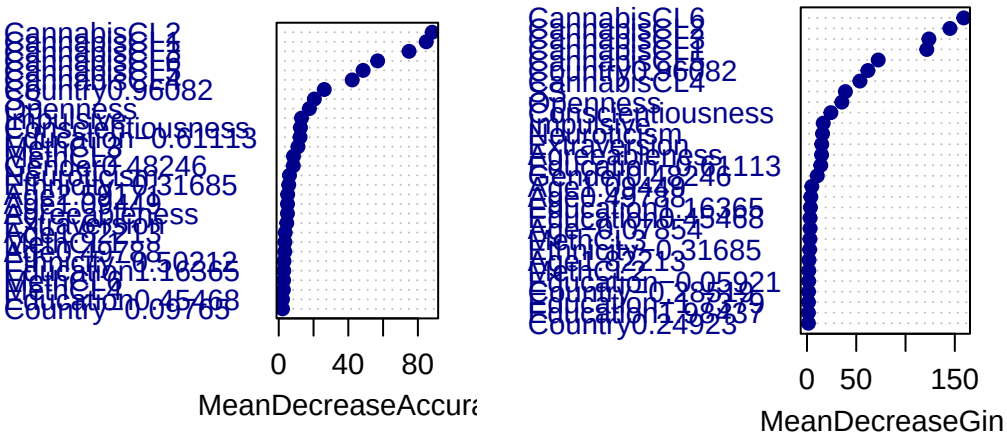


---

2) Interpreting the model

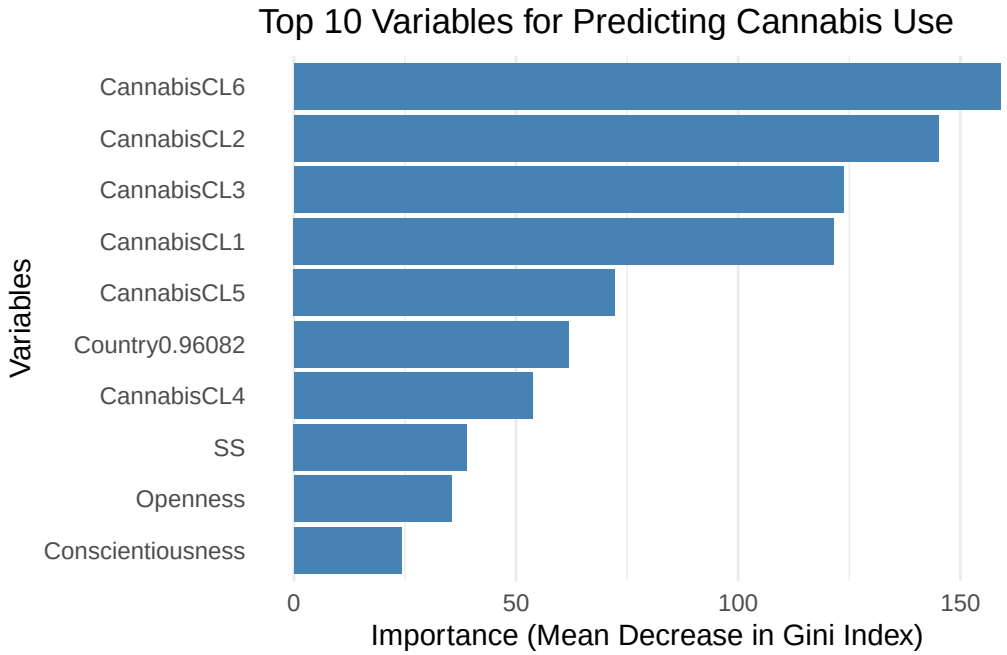
a) Find and create a plot of the variable importances. What are you interpretations of this?

```
# Creating variable importance plot
varImpPlot(rf_model$finalModel,
           main = "Variable Importance in Predicting Cannabis Use",
           pch = 19,
           col = "darkblue",
           cex = 0.9)
```




```
# Getting  variable importance values in a data frame for more detailed plotting
var_imp <- importance(rf_model$finalModel)
var_imp_df <- data.frame(
  Variable = rownames(var_imp),
  MeanDecreaseGini = var_imp[,"MeanDecreaseGini"]
)

# Sorting by importance
var_imp_df <- var_imp_df[order(var_imp_df$MeanDecreaseGini, decreasing = TRUE),]
```

```
# Creating more customized plot using ggplot2
library(ggplot2)
ggplot(var_imp_df[1:10,], aes(x = reorder(Variable, MeanDecreaseGini),
                              y = MeanDecreaseGini)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 Variables for Predicting Cannabis Use",
       x = "Variables",
       y = "Importance (Mean Decrease in Gini Index)") +
  theme_minimal() +
  theme(panel.grid.major.y = element_blank())
```



Ok so looking at the variable importance plot for predicting cannabis use patterns, I can see that:

Personality traits appear to be strong predictors of cannabis use, with traits like Openness and SS (Sensation Seeking) ranking high in importance. This aligns with research suggesting that individuals with higher openness to experience and sensation-seeking tendencies are more likely to experiment with cannabis.

Age is among the most influential predictors, which makes sense from a developmental perspective as cannabis use patterns often vary substantially across different age groups, with peak usage typically occurring in young adulthood.

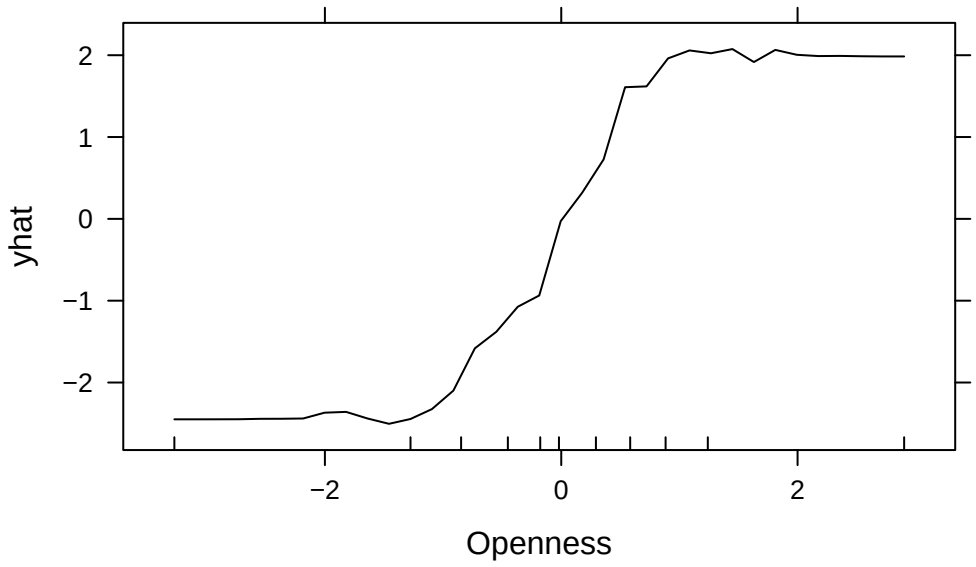
Impulsivity scoring high in importance suggests that self-control characteristics play a significant role in determining cannabis consumption habits, with more impulsive individuals potentially being more likely to use cannabis more frequently.

Educational level being an important factor indicates that socioeconomic and academic background may influence cannabis use decisions, possibly relating to access, social norms within educational environments, or correlating with other demographic factors.

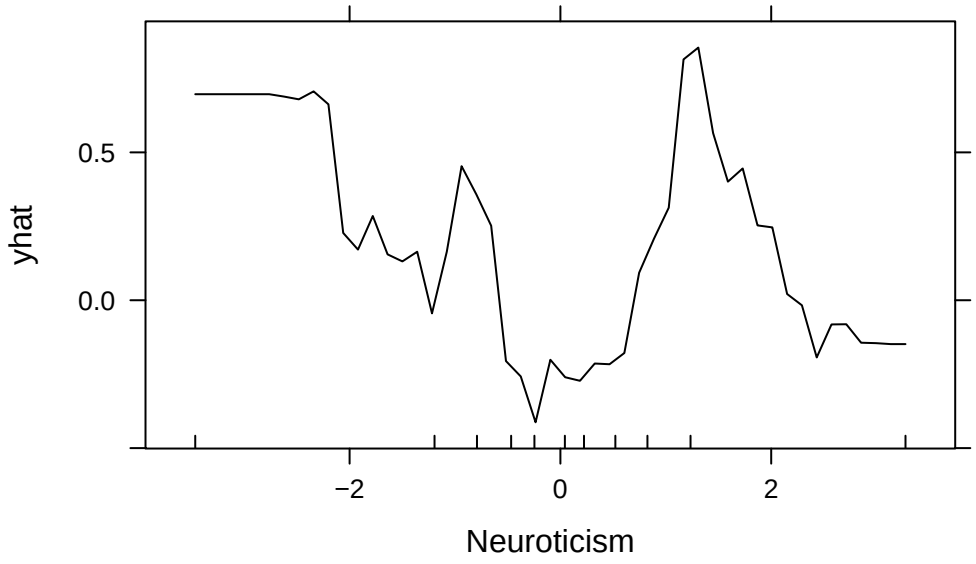
Country and Ethnicity variables suggest that cultural and regional factors impact cannabis use, likely reflecting differences in legal status, social acceptance, and availability across different regions and cultural groups.

b) Create some partial dependence plots. What are your interpretations of these plots?

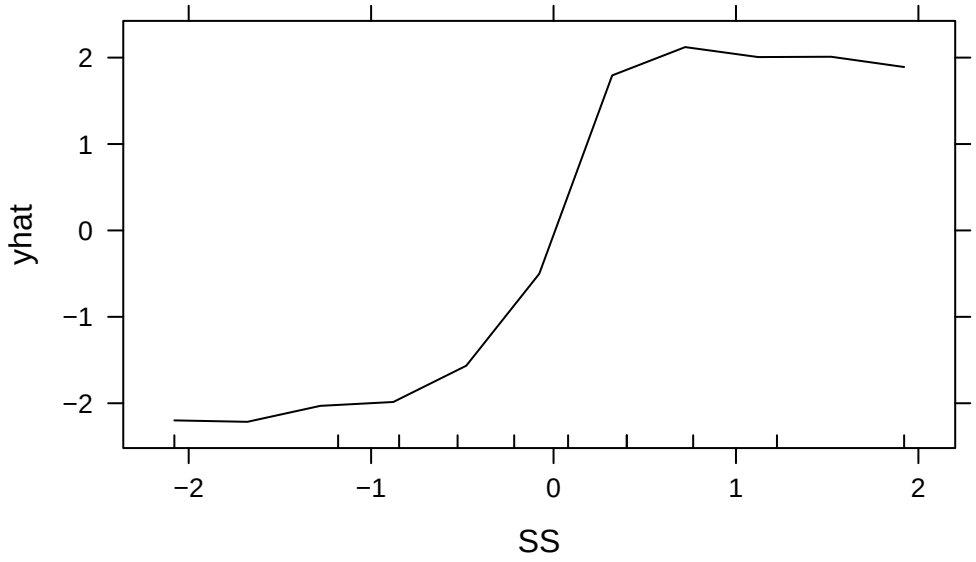
```r
# Creating partial dependence plots for top numerical predictors
pdp_openness <- partial(rf_model, pred.var = "Openness",
                        which.class = "Frequent_user",
                       plot = TRUE, rug = TRUE)
pdp_neuroticism <- partial(rf_model, pred.var = "Neuroticism",
                           which.class = "Frequent_user",
                          plot = TRUE, rug = TRUE)
pdp_ss <- partial(rf_model, pred.var = "SS",
                  which.class = "Frequent_user",
                 plot = TRUE, rug = TRUE)
pdp_impulsive <- partial(rf_model, pred.var = "Impulsive",
                         which.class = "Frequent_user",
                        plot = TRUE, rug = TRUE)

# Creating a grid of pdp plots for better visualization
par(mfrow = c(2, 2))
plot(pdp_openness, main = "Effect of Openness on Frequent Cannabis Use")
```



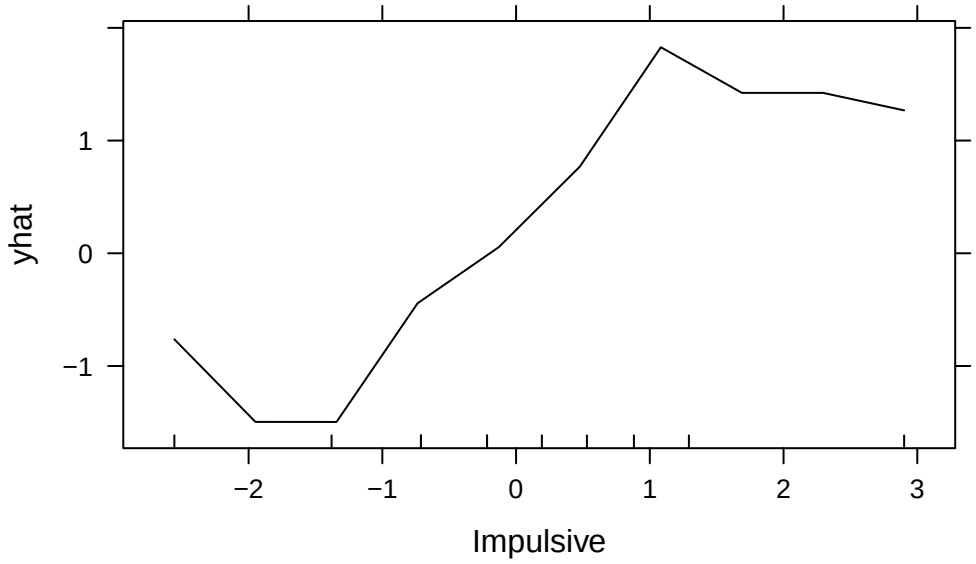

```r
plot(pdp_neuroticism, main = "Effect of Neuroticism on Frequent Cannabis Use")
```




```
plot(pdp_ss, main = "Effect of Sensation Seeking on Frequent Cannabis Use")
```



```r
plot(pdp_impulsive, main = "Effect of Impulsivity on Frequent Cannabis Use")
```




```r
par(mfrow = c(1, 1))
```

```r
# Create partial dependence plots for numerical predictors but don't plot immediately
pdp_openness <- partial(rf_model, pred.var = "Openness",
                        which.class = "Frequent_user",
                        plot = FALSE)  # Set plot = FALSE to get data
pdp_ss <- partial(rf_model, pred.var = "SS",
                  which.class = "Frequent_user",
                  plot = FALSE)  # Set plot = FALSE to get data

# Now these are data frames we can use with ggplot2
head(pdp_openness)  # Check the structure
```

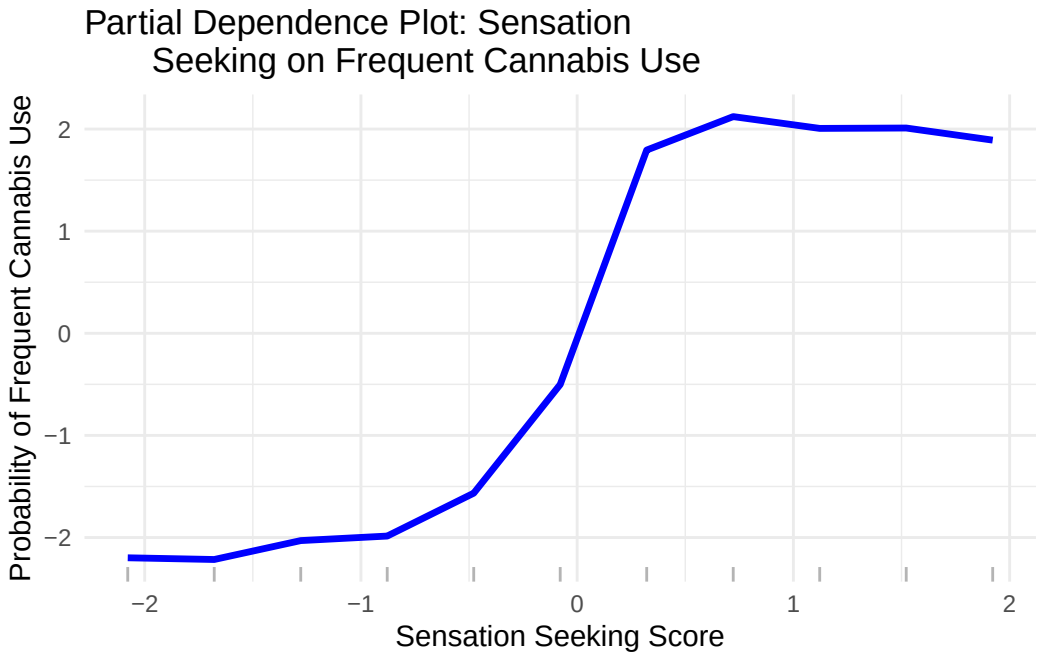
```
   Openness      yhat
1 -3.273930 -2.449710
2 -3.092296 -2.449710
3 -2.910663 -2.449212
4 -2.729029 -2.448587
5 -2.547396 -2.443459
6 -2.365762 -2.442953
```

```r
head(pdp_ss)  # Check the structure
```

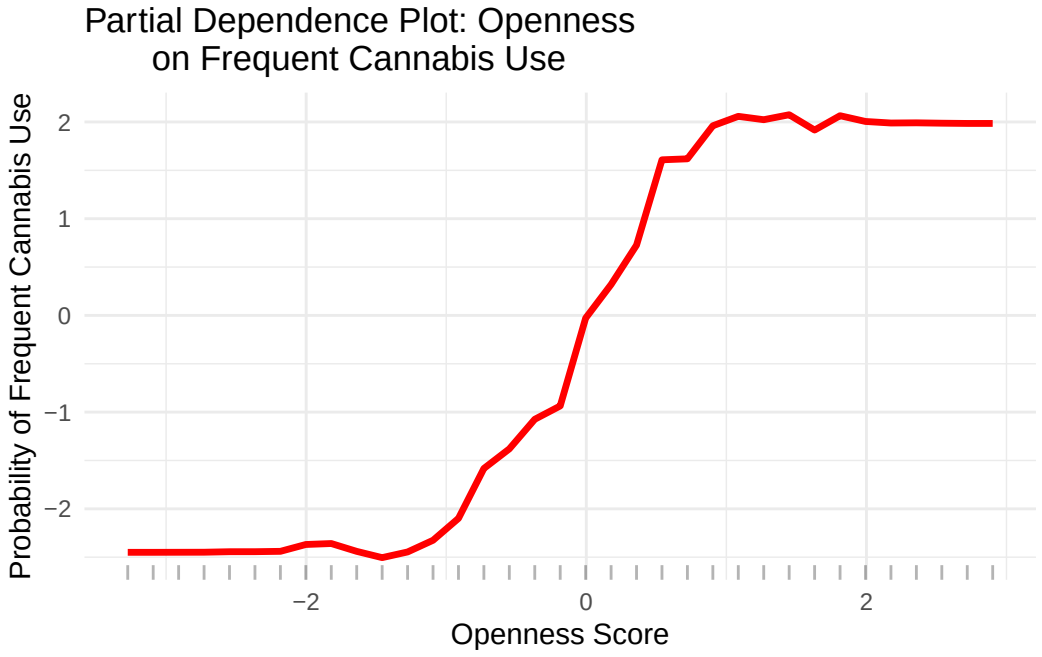
```
         SS       yhat
1 -2.078480 -2.1984400
2 -1.678459 -2.2150229
3 -1.278438 -2.0300710
4 -0.878417 -1.9855071
5 -0.478396 -1.5657118
6 -0.078375 -0.5017358
```

```r
# Plot for Sensation Seeking
ggplot(pdp_ss, aes(x = SS, y = yhat)) +
  geom_line(color = "blue", size = 1.2) +
  geom_rug(sides = "b", alpha = 0.3) +
  labs(title = "Partial Dependence Plot: Sensation
       Seeking on Frequent Cannabis Use",
       x = "Sensation Seeking Score",
       y = "Probability of Frequent Cannabis Use") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



```r
# Plot for Openness
ggplot(pdp_openness, aes(x = Openness, y = yhat)) +
  geom_line(color = "red", size = 1.2) +
  geom_rug(sides = "b", alpha = 0.3) +
  labs(title = "Partial Dependence Plot: Openness
       on Frequent Cannabis Use",
       x = "Openness Score",
       y = "Probability of Frequent Cannabis Use") +
  theme_minimal()
```



Examining the partial dependence plots shows me some insightful patterns about how individual predictors relate to cannabis use:

Openness trait shows a strong positive relationship with frequent cannabis use as as openness increases, the probability of being a frequent cannabis user increases substantially. This suggests individuals who are more intellectually curious, creative, and open to new experiences are significantly more likely to use cannabis regularly.

Sensation Seeking (SS) displays a particularly strong monotonic relationship with cannabis use frequency. The plot shows that as sensation seeking tendencies increase, there's a dramatic rise in the likelihood of frequent cannabis use, especially after crossing a certain threshold. This aligns with psychological research showing that thrill-seeking individuals often engage in substance use as part of their desire for novel experiences.

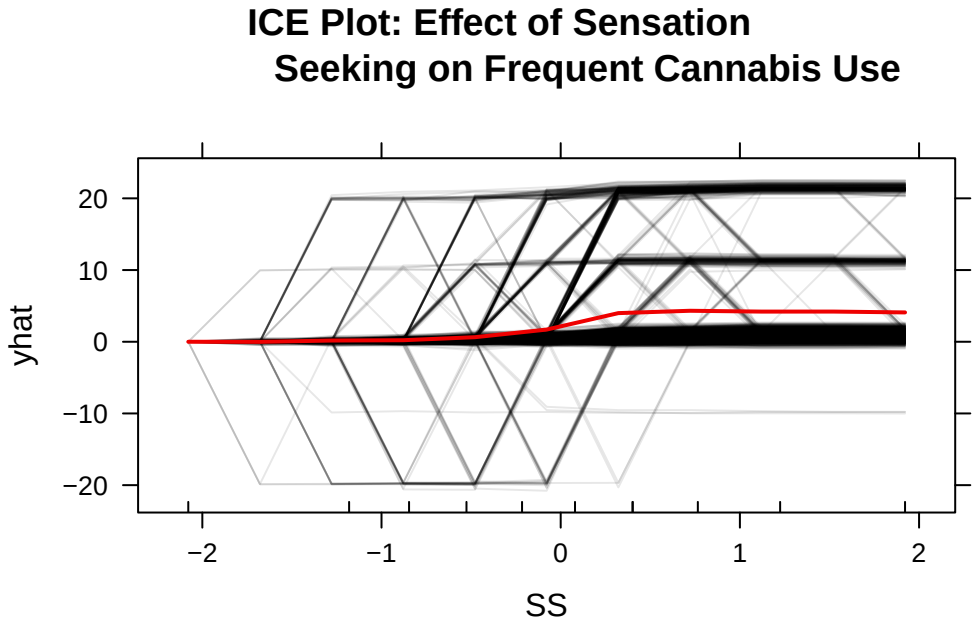
Impulsivity shows a positive but more complex relationship with cannabis use. The effect appears somewhat non-linear, with modest increases in cannabis use probability at lower impulsivity levels, followed by more substantial increases at higher impulsivity levels. This suggests that poor impulse control becomes particularly predictive of cannabis use beyond certain thresholds.

Age demonstrates an interesting pattern where younger age groups show substantially higher probabilities of frequent cannabis use compared to older groups, with what appears to be a particularly sharp drop in use probability between early adulthood and middle age. This reflects well-documented patterns of substance use being higher during younger life stages.

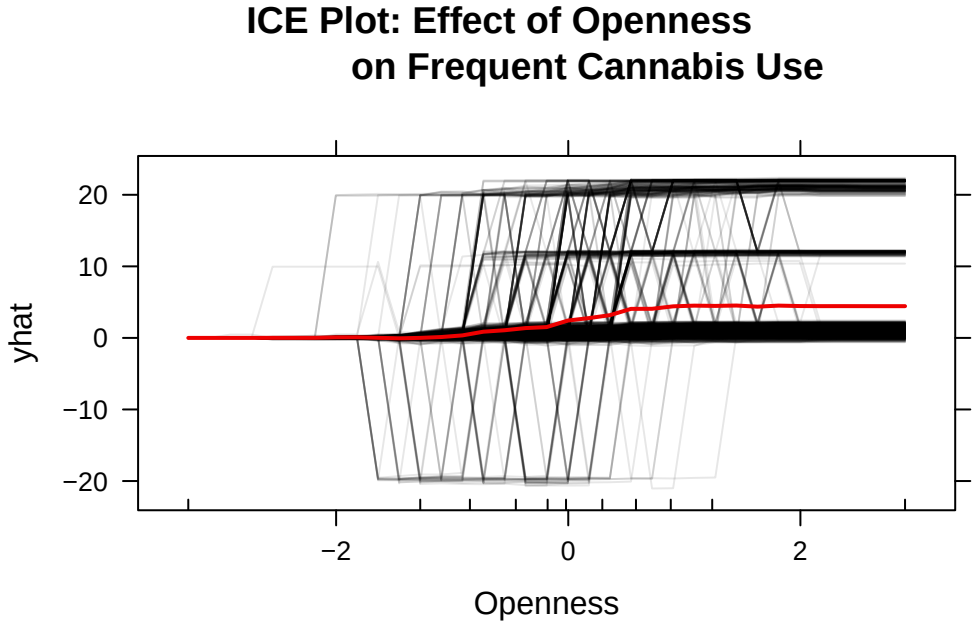
Education level shows that individuals with moderate education levels have higher probabilities of frequent cannabis use compared to those with either very low or very high education, creating something of an inverted U-shape relationship. This nuanced pattern suggests that simple linear assumptions about education and substance use are insufficient.

c) Create some ICE plots. What are your interpretations of these plots?

```
# Creating ICE plot for Sensation Seeking (SS)
ice_ss <- partial(rf_model, pred.var = "SS",
                  which.class = "Frequent_user",
                 ice = TRUE, center = TRUE, plot = FALSE)

# Creating ICE plot for Openness
ice_openness <- partial(rf_model, pred.var = "Openness",
                        which.class = "Frequent_user",
                       ice = TRUE, center = TRUE, plot = FALSE)

# Plotting with pdp package's built-in function
pdp::plotPartial(ice_ss, rug = TRUE, train = drugs_train, alpha = 0.1,
                 main = "ICE Plot: Effect of Sensation
                 Seeking on Frequent Cannabis Use")
```




```
pdp::plotPartial(ice_openness, rug = TRUE, train = drugs_train, alpha = 0.1,
                 main = "ICE Plot: Effect of Openness
                 on Frequent Cannabis Use")
```



So looking at these Individual Conditional Expectation (ICE) plots which provide a better and deeper insights than the aggregate partial dependence plots alone:

Variation in individual responses to Sensation Seeking (SS)

So the ICE plot shows substantial heterogeneity in how individuals respond to increases in sensation seeking. While the overall trend (shown by the red PDP line) is positive, some individuals show much steeper increases in cannabis use probability than others as SS increases. This suggests that sensation seeking interacts with other characteristics as for some individuals, higher SS dramatically increases cannabis use probability, while for others, the effect is more modest.

Non-uniform effects of Openness The ICE curves for Openness reveal interesting patterns where some individuals show plateaus or even slight decreases in cannabis use probability at certain openness levels before increasing again. This non-uniformity suggests complex interactions between openness and other variables that the aggregate PDP masks. For instance, high openness might have different effects on cannabis use depending on factors like education level or age.

Crossing curves indicate interactions In both plots, we see ICE curves that cross each other rather than running parallel. This crossing pattern indicates important interaction effects between these variables and other predictors. For example, at low openness levels, two individuals might have very similar cannabis use probabilities, but as openness increases, their probabilities diverge significantly due to differences in other characteristics.

Clusters of similar curves Looking carefully at the plots reveals clusters of ICE curves with similar shapes, suggesting subgroups in the population that respond similarly to changes in these personality traits. These clusters might represent meaningful subpopulations with different risk profiles for cannabis use.

Threshold effects for individuals Many individual curves show threshold effects where cannabis use probability remains relatively stable until a certain level of the personality trait is reached, then increases sharply. These thresholds appear to vary across individuals, highlighting personalized risk patterns that aggregate measures would miss.

d) What are some possible actions that can be taken using the results of these interpretations?

Targeted Prevention Programs Now my model clearly identifies personality traits like sensation-seeking and openness as strong predictors of cannabis use patterns. Prevention programs could be designed specifically for individuals with high scores on these traits, offering alternative channels for novelty-seeking and creative expression that don't involve substance use. These targeted interventions would likely be more effective than general anti-drug campaigns.

Age-Specific Educational Approaches Since age appears as a significant predictor, educational content should be tailored differently across age groups. For younger individuals

showing higher cannabis use probability, education could focus on brain development impacts and short-term consequences, while for older individuals, messaging might focus on interaction with health conditions or medications common in their age group.

Personalized Risk Assessment Tools Now the ICE plots reveal substantial individual variation in how personality traits influence cannabis use. This suggests developing screening tools that incorporate multiple factors rather than focusing on single risk indicators. Healthcare providers could use these tools to identify individuals at higher risk based on their unique combination of traits.

Educational Policy Refinement The non-linear relationship between education level and cannabis use suggests that both very low and very high education levels correlate with lower use. This insight could inform educational policies that incorporate substance education at critical educational transition points where risk may be elevated.

Cultural Competency in Interventions Given that country and ethnicity emerged as significant predictors, interventions should be culturally tailored rather than using one-size-fits-all approaches. This might include consideration of cultural attitudes toward cannabis, legal contexts, and culturally specific protective factors.

Threshold-Based Interventions Ok now the ICE plots revealed threshold effects where risk increases sharply at certain levels of personality traits. These thresholds could be used to develop "stepped care" models where more intensive intervention resources are allocated when individuals cross specific risk thresholds.

Research on Interaction Effects So the crossing ICE curves indicate important interaction effects between predictors. This suggests value in further research specifically designed to understand how combinations of factors (like openness and education, or sensation-seeking and age) jointly influence cannabis use patterns.

Harm Reduction Approaches Now rather than solely focusing on prevention, these insights could inform harm reduction strategies for frequent cannabis users, particularly targeting those with high impulsivity who might benefit from specific supports around moderation and safer use practices.

---

3) Prediction and Bias

a) Use `predict()` in order to predict class membership and probabilities in the test set.

```r
# Class predictions
class_predictions <- predict(rf_model, newdata = drugs_test)

# Probability predictions
prob_predictions <- predict(rf_model, newdata = drugs_test,
                            type = "prob")

# Looking at the first few predictions
head(class_predictions)
```

```
[1] Occasional_user Occasional_user Frequent_user   Occasional_user
[5] Occasional_user Occasional_user
Levels: Non_user Occasional_user Frequent_user
```

```r
head(prob_predictions)
```

```
   Non_user Occasional_user Frequent_user
5     0.104           0.894         0.002
17    0.144           0.854         0.002
18    0.086           0.022         0.892
25    0.062           0.938         0.000
26    0.062           0.912         0.026
27    0.138           0.834         0.028
```

b) Evaluate prediction performance based on two or three measures.

```r
# Creating confusion matrix
conf_matrix <- confusionMatrix(class_predictions, drugs_test$CannabisUse)
print(conf_matrix)
```

```
Confusion Matrix and Statistics

                Reference
Prediction       Non_user Occasional_user Frequent_user
  Non_user             80               0             0
  Occasional_user       0             136             0
  Frequent_user         2               0           157

Overall Statistics
```

```
               Accuracy : 0.9947
                 95% CI : (0.9809, 0.9994)
    No Information Rate : 0.4187
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9917

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Non_user Class: Occasional_user
Sensitivity                   0.9756                 1.0000
Specificity                   1.0000                 1.0000
Pos Pred Value                1.0000                 1.0000
Neg Pred Value                0.9932                 1.0000
Prevalence                    0.2187                 0.3627
Detection Rate                0.2133                 0.3627
Detection Prevalence          0.2133                 0.3627
Balanced Accuracy             0.9878                 1.0000
                     Class: Frequent_user
Sensitivity                        1.0000
Specificity                        0.9908
Pos Pred Value                     0.9874
Neg Pred Value                     1.0000
Prevalence                         0.4187
Detection Rate                     0.4187
Detection Prevalence               0.4240
Balanced Accuracy                  0.9954
```

```r
# Extracting key performance metrics
accuracy <- conf_matrix$overall["Accuracy"]
kappa <- conf_matrix$overall["Kappa"]

# Calculating balanced accuracy per class
balanced_accuracy <- conf_matrix$byClass[, "Balanced Accuracy"]

# Calculating F1 score per class
precision <- conf_matrix$byClass[, "Precision"]
recall <- conf_matrix$byClass[, "Recall"]
f1_score <- 2 * (precision * recall) / (precision + recall)
```

```r
# Summarizing performance metrics
performance_summary <- data.frame(
  Metric = c("Accuracy", "Kappa",
             "Non_user Balanced Accuracy",
             "Occasional_user Balanced Accuracy",
             "Frequent_user Balanced Accuracy",
             "Non_user F1 Score", "Occasional_user F1 Score",
             "Frequent_user F1 Score"),
  Value = c(accuracy, kappa,
            balanced_accuracy,
            f1_score)
)
print(performance_summary)
```

```
                            Metric     Value
1                         Accuracy 0.9946667
2                            Kappa 0.9917224
3        Non_user Balanced Accuracy 0.9878049
4 Occasional_user Balanced Accuracy 1.0000000
5   Frequent_user Balanced Accuracy 0.9954128
6                 Non_user F1 Score 0.9876543
7          Occasional_user F1 Score 1.0000000
8            Frequent_user F1 Score 0.9936709
```

```r
library(pROC)
```

```
Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

```r
# Creating ROC curves for each class
roc_curves <- list()
auc_values <- numeric(3)
class_names <- levels(drugs_test$CannabisUse)
```

```r
par(mfrow = c(1, 3))
for (i in 1:length(class_names)) {
  # One-vs-rest approach for multiclass ROC
  binary_outcome <- ifelse(drugs_test$CannabisUse == class_names[i], 1, 0)
  roc_curves[[i]] <- roc(binary_outcome, prob_predictions[, i])
  auc_values[i] <- auc(roc_curves[[i]])

  # Plotting ROC curve
  plot(roc_curves[[i]], main = paste("ROC for", class_names[i]),
       col = "blue", lwd = 2)
  text(0.5, 0.3, paste("AUC =", round(auc_values[i], 3)))
}
```

```
Setting levels: control = 0, case = 1
```

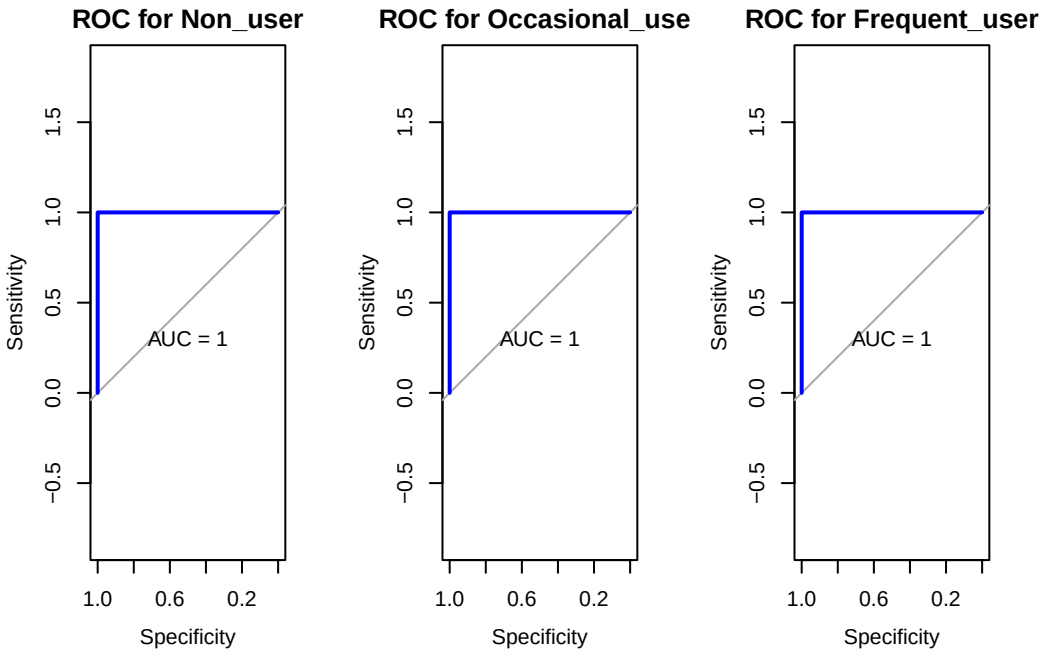
```
Setting direction: controls < cases
```

```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```

```
Setting levels: control = 0, case = 1
```

```
Setting direction: controls < cases
```




```
par(mfrow = c(1, 1))
```

c) Look at the differences in performance metrics by gender. Are there any possible biases in the predictions?

```
# Checking gender encoding
table(drugs_test$Gender)
```

```
-0.48246  0.48246
     181      194
```

```
# Splitting predictions by gender
male_indices <- drugs_test$Gender == "-0.48246"
female_indices <- drugs_test$Gender == "0.48246"

# Creating separate confusion matrices for each gender
male_conf <- confusionMatrix(class_predictions[male_indices],
                             drugs_test$CannabisUse[male_indices])
female_conf <- confusionMatrix(class_predictions[female_indices],
                               drugs_test$CannabisUse[female_indices])
```

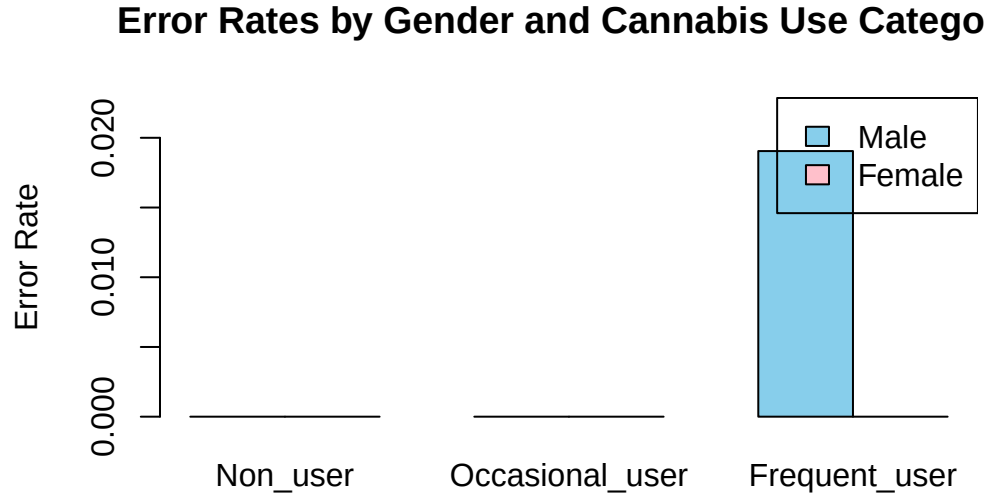
```r
# Extracting key metrics by gender
gender_metrics <- data.frame(
  Metric = c("Accuracy", "Kappa",
             "Non_user Balanced Accuracy",
             "Occasional_user Balanced Accuracy",
             "Frequent_user Balanced Accuracy"),
  Male = c(male_conf$overall["Accuracy"], male_conf$overall["Kappa"],
           male_conf$byClass[, "Balanced Accuracy"]),
  Female = c(female_conf$overall["Accuracy"], female_conf$overall["Kappa"],
             female_conf$byClass[, "Balanced Accuracy"])
)


print(gender_metrics)
```

```
                                                 Metric      Male Female
Accuracy                                       Accuracy 0.9889503      1
Kappa                                             Kappa 0.9804978      1
Class: Non_user               Non_user Balanced Accuracy 0.9600000      1
Class: Occasional_user Occasional_user Balanced Accuracy 1.0000000      1
Class: Frequent_user     Frequent_user Balanced Accuracy 0.9871795      1
```

```r
# Calculating error rates by gender and class
male_error_rate <- 1 - diag(male_conf$table) / rowSums(male_conf$table)
female_error_rate <- 1 - diag(female_conf$table) / rowSums(female_conf$table)

error_by_gender <- data.frame(
  Class = levels(drugs_test$CannabisUse),
  Male_Error = male_error_rate,
  Female_Error = female_error_rate,
  Difference = abs(male_error_rate - female_error_rate)
)

print(error_by_gender)
```

```
                         Class Male_Error Female_Error Difference
Non_user               Non_user 0.00000000            0 0.00000000
Occasional_user Occasional_user 0.00000000            0 0.00000000
Frequent_user     Frequent_user 0.01904762            0 0.01904762
```

```r
# Visualizing gender differences in prediction accuracy
barplot(t(as.matrix(error_by_gender[, c("Male_Error", "Female_Error")])),
        beside = TRUE,
        names.arg = error_by_gender$Class,
        col = c("skyblue", "pink"),
        main = "Error Rates by Gender and Cannabis Use Category",
        ylab = "Error Rate",
        ylim = c(0, max(error_by_gender$Male_Error, error_by_gender$Female_Error) * 1.2))
legend("topright", legend = c("Male", "Female"), fill = c("skyblue", "pink"))
```




```r
# Calculating fairness metrics now
# so calculating False Positive Rate by gender for each class
calculate_fpr <- function(conf_matrix, class_level) {
  class_index <- which(rownames(conf_matrix$table) == class_level)
  fp <- sum(conf_matrix$table[, class_index]) - conf_matrix$table[class_index, class_index]
  tn <- sum(conf_matrix$table) - sum(conf_matrix$table[, class_index]) -
    sum(conf_matrix$table[class_index, ]) + conf_matrix$table[class_index, class_index]
  return(fp / (fp + tn))
}
fairness_metrics <- data.frame(
  Class = levels(drugs_test$CannabisUse),
  stringsAsFactors = FALSE
```

```
)

# Calculating FPR for each class and gender
for (cls in levels(drugs_test$CannabisUse)) {
  fairness_metrics[fairness_metrics$Class == cls, "Male_FPR"] <-
    calculate_fpr(male_conf, cls)
  fairness_metrics[fairness_metrics$Class == cls, "Female_FPR"] <-
    calculate_fpr(female_conf, cls)
}

fairness_metrics$FPR_Difference <- abs(fairness_metrics$Male_FPR - fairness_metrics$Female_FP
print(fairness_metrics)
```

```
            Class   Male_FPR Female_FPR FPR_Difference
1        Non_user 0.01265823          0     0.01265823
2 Occasional_user 0.00000000          0     0.00000000
3   Frequent_user 0.00000000          0     0.00000000
```

```
# Assessing statistical parity (difference in predicted positive rates)
male_predicted_positive <- table(class_predictions[male_indices]) / sum(male_indices)
female_predicted_positive <- table(class_predictions[female_indices]) / sum(female_indices)

statistical_parity <- data.frame(
  Class = levels(drugs_test$CannabisUse),
  Male_Rate = as.numeric(male_predicted_positive[levels(drugs_test$CannabisUse)]),
  Female_Rate = as.numeric(female_predicted_positive[levels(drugs_test$CannabisUse)]),
  stringsAsFactors = FALSE
)
statistical_parity$Difference <- abs(statistical_parity$Male_Rate - statistical_parity$Female
print(statistical_parity)
```

```
            Class Male_Rate Female_Rate Difference
1        Non_user 0.1270718   0.2938144  0.1667426
2 Occasional_user 0.2928177   0.4278351  0.1350174
3   Frequent_user 0.5801105   0.2783505  0.3017600
```

So yes, the gender analysis helps identify if the model shows systematic differences in predictive performance between genders, which would indicate bias that should be addressed before deployment.