Assignment 5: Interpretation and Bias Considerations in ML

Cheng, Chia Wen

2023-04-10

Setup

```
library(caret)
library(randomForest)
library(partykit)
library(pdp)
library(iml)
library(fastAdaboost)
library(dplyr)
library(MLmetrics)
library(ranger)
library(ada)
```

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.

```
probability <- drugs %>% group_by(LSD) %>% summarise(Percentage=n()/nrow(.))
length(drugs$LSD)
```

```
## [1] 1885
```

```
table(drugs$LSD)
```

```
##
##  CL0  CL1  CL2  CL3  CL4  CL5  CL6
## 1069  259  177  214   97   56   13
```

```
# is.na(drugs$LSD) # no missing value
drugs$LSD <- as.factor(drugs$LSD)
```

I'm using LSD as my outcome variable. There are 1885 data in this variable without missing values. It is a categorical variable with 7 classes. CL0 (56.71%) means never used; CL1 (13.74%) means used over a decade ago; CL2 (9.39%) means used in last decade; CL3 (11.35%) means used in last year; CL4 (5.15%) means used in last month; CL5 (2.97%) means used in last week; and CL6 (0.69%) means used in last day.

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

```
set.seed(9574)

inTrain <- createDataPartition(drugs$LSD,
                               p = .8,
                               list = FALSE,
                               times = 1)

drugs_train <- drugs[inTrain,]
drugs_test <- drugs[-inTrain,]
drugs_train %>% group_by(LSD) %>% summarise(Percentage=n()/nrow(.))
```

```
## # A tibble: 7 x 2
##   LSD   Percentage
##   <fct>      <dbl>
## 1 CL0      0.566
## 2 CL1      0.138
## 3 CL2      0.0939
## 4 CL3      0.114
## 5 CL4      0.0516
## 6 CL5      0.0298
## 7 CL6      0.00728
```

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

```
ctrl  <- trainControl(method = "cv",
                      number = 10,
                      summaryFunction = multiClassSummary,
                      classProbs = TRUE)
```

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

```
grid <- expand.grid(mtry = c(sqrt(ncol(drugs_train)), # number randomly variable selected is mtry
                             log(ncol(drugs_train))),
                    splitrule = c("gini"),
                    min.node.size = 10)
grid
```

```
##       mtry splitrule min.node.size
## 1 5.656854      gini            10
## 2 3.465736      gini            10
```

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.

```
set.seed(9574)
rf <- train(LSD ~ Age + Gender + Education + Country + Ethnicity + Neuroticism + Extraversion + Openness
            data = drugs_train,
            method = "ranger",
            trControl = ctrl,
            tuneGrid = grid,
            metric = "ROC",
            importance = "impurity")
rf
```

```
## Random Forest
##
## 1512 samples
##   12 predictor
##    7 classes: 'CL0', 'CL1', 'CL2', 'CL3', 'CL4', 'CL5', 'CL6'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1361, 1359, 1362, 1360, 1362, 1362, ...
## Resampling results across tuning parameters:
##
##   mtry      logLoss   AUC        prAUC      Accuracy   Kappa       Mean_F1
##   3.465736  1.109108  0.8133527  0.2900101  0.5780107  0.09660001  NaN
##   5.656854  1.099855  0.8074784  0.2876314  0.6018833  0.20587919  NaN
##   Mean_Sensitivity  Mean_Specificity  Mean_Pos_Pred_Value  Mean_Neg_Pred_Value
##   0.1700493         0.8692273         NaN                  0.9109249
##   0.2115019         0.8837558         NaN                  0.9194463
##   Mean_Precision  Mean_Recall  Mean_Detection_Rate  Mean_Balanced_Accuracy
##   NaN             0.1700493    0.08257295           0.5196383
##   NaN             0.2115019    0.08598332           0.5476289
##
## Tuning parameter 'splitrule' was held constant at a value of gini
##
## Tuning parameter 'min.node.size' was held constant at a value of 10
## logLoss was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 5.656854, splitrule = gini
##  and min.node.size = 10.
```

```
#treeInfo(rf$finalModel)
```

If we'd like a greater AUC to better distinguish positive and negative cases, the best model is the tree built with 'mtry' at 3.465736. If we're interested in a higher accuracy that the overall correctness of the model is higher, the best model is the tree built with 'mtry' at 5.656854.

---

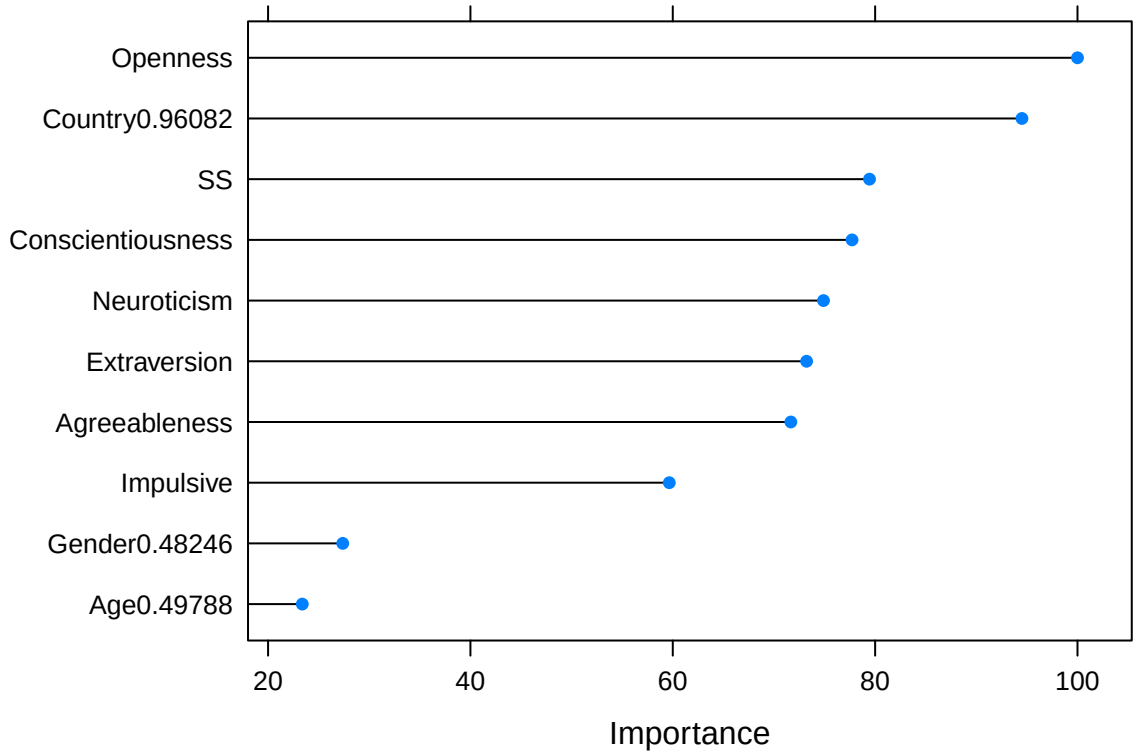
2) Interpreting the model

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

```
varImp(rf, scale = FALSE)
```

```
## ranger variable importance
##
##   only 20 most important variables shown (out of 33)
##
##                    Overall
## Openness           63.735
## Country0.96082     60.275
## SS                 50.785
## Conscientiousness  49.696
## Neuroticism        47.918
## Extraversion       46.868
## Agreeableness      45.887
## Impulsive          38.318
## Gender0.48246      17.985
## Age0.49788         15.466
## Age1.09449         15.281
## Education-0.61113  13.174
## Age-0.07854        10.776
## Education0.45468    8.986
## Age1.82213          8.197
## Country-0.28519     6.532
## Education-0.05921   6.395
## Education1.16365    6.133
## Country0.24923      5.674
## Country-0.09765     5.387
```

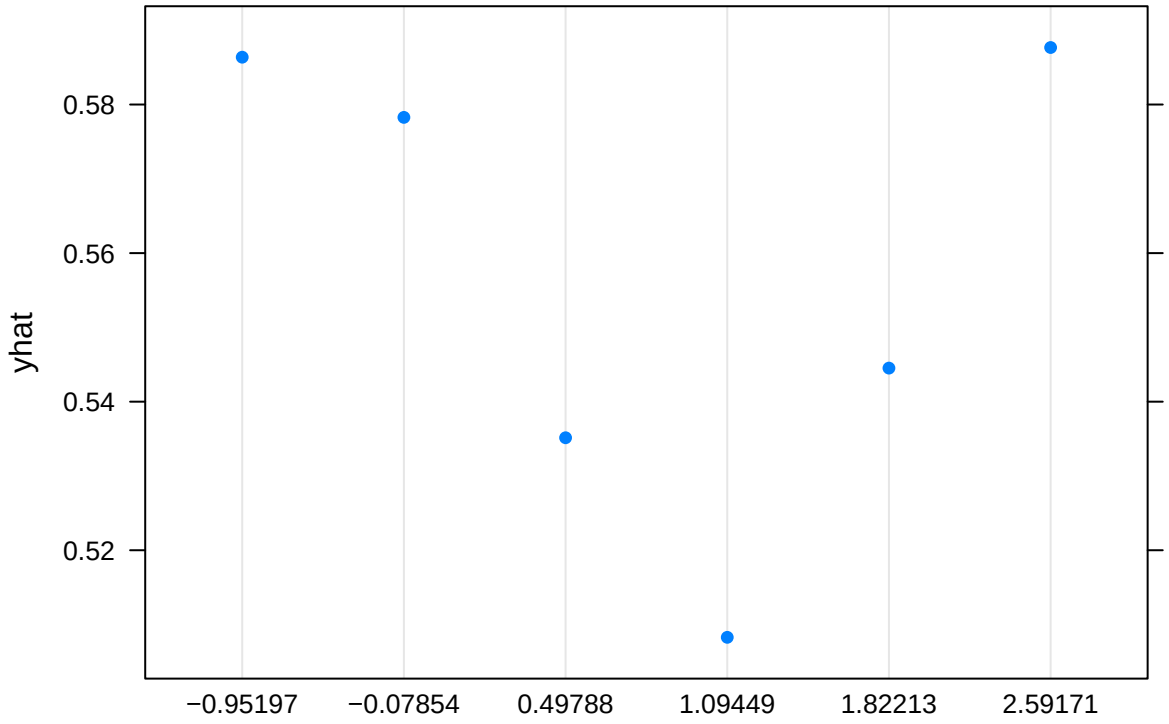
```
plot(varImp(rf), top = 10)
```



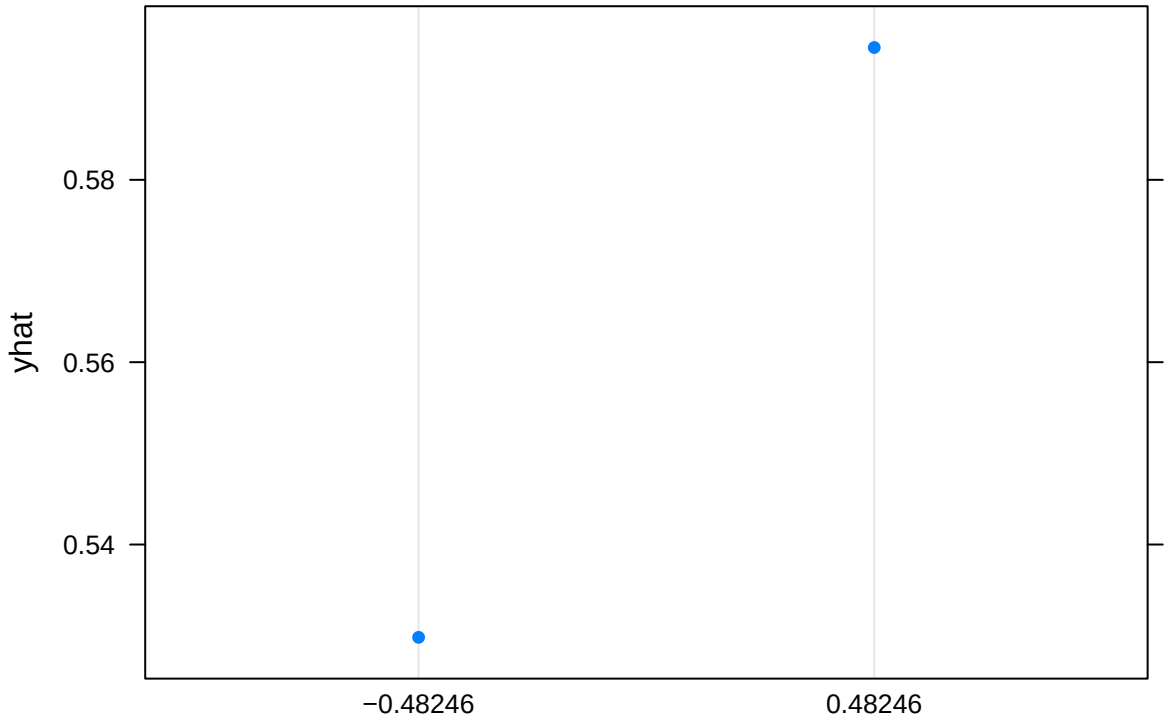
"When using the Gini index as impurity function, this measure is known as the Gini importance or Mean Decrease Gini." From the plot, 'Openness' has an importance on top of all the other variables, followed by 'Country' of 0.96082, 'SS,' 'Conscientiousness,' 'Neuroticism,' 'Extraversion,' 'Agreeableness,' and 'Impulsive.' However, the results of variable importance do nothing with relationships between the variables.

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

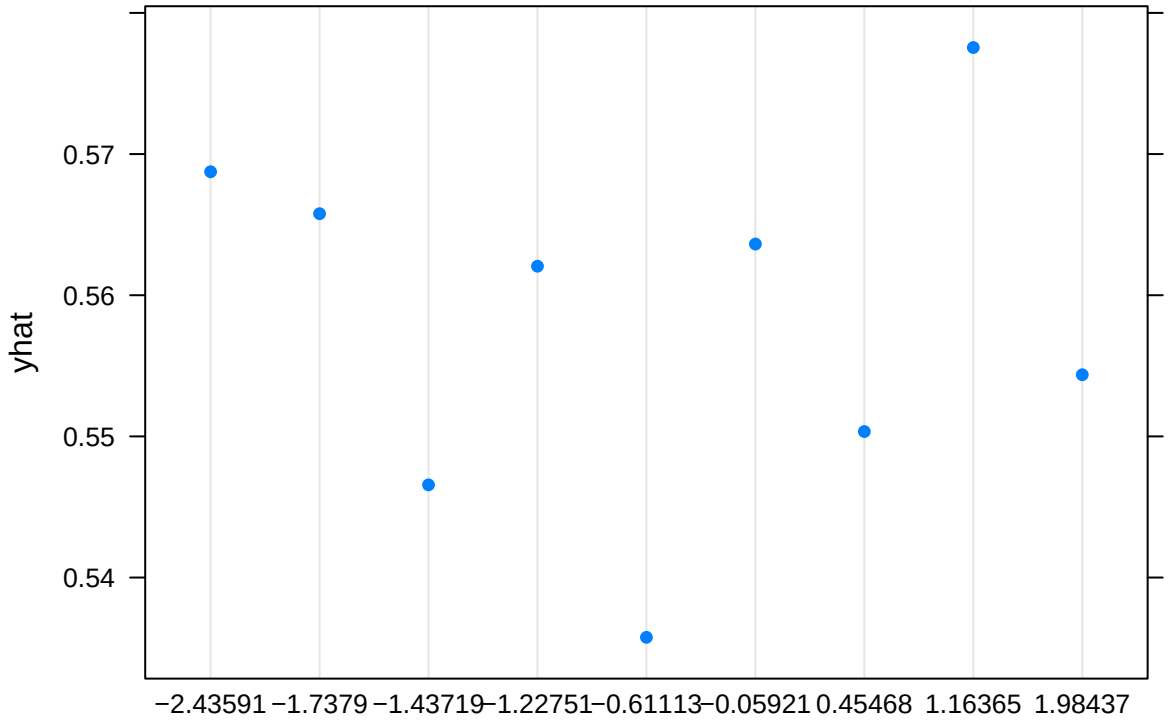
```
pdp1 <- partial(rf, pred.var = "Age",
                type = "classification",
                which.class = 1, prob = T)
p1 <- plotPartial(pdp1, rug = T, train = drugs_train)
p1
```



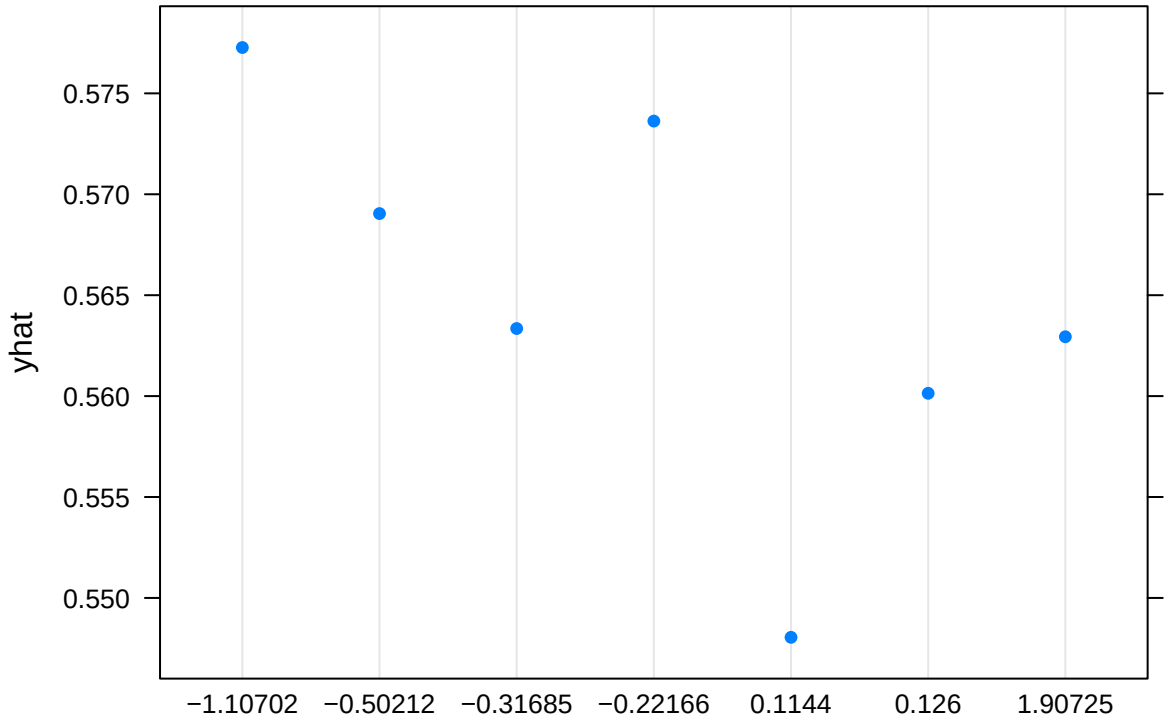

```
pdp2 <- partial(rf, pred.var = "Gender",
                type = "classification",
                which.class = 1, prob = T)
p2 <- plotPartial(pdp2, rug = T, train = drugs_train)
p2
```



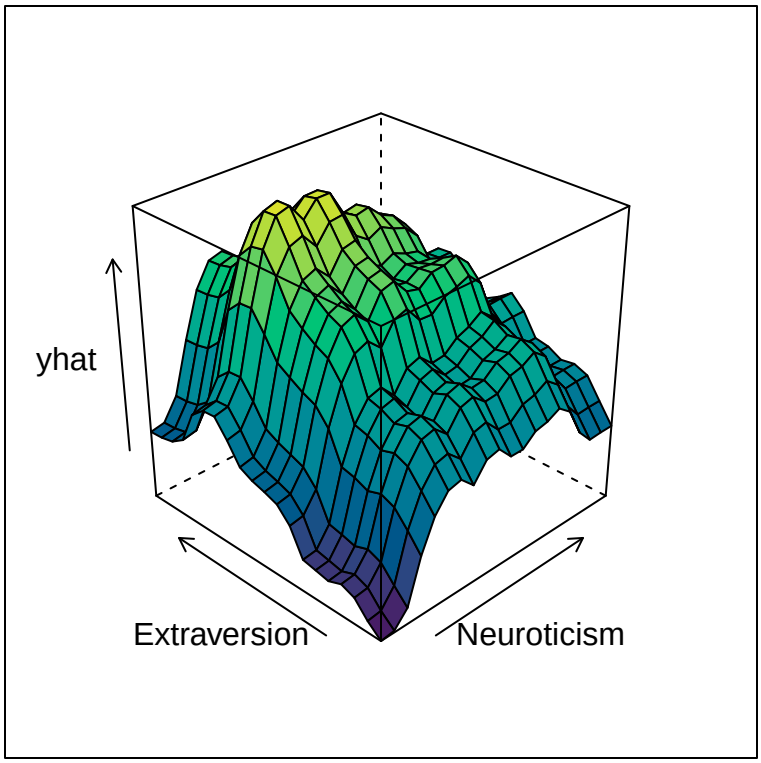

```
pdp3 <- partial(rf, pred.var = "Education",
                type = "classification",
                which.class = 1, prob = T)
p3 <- plotPartial(pdp3, rug = T, train = drugs_train)
p3
```



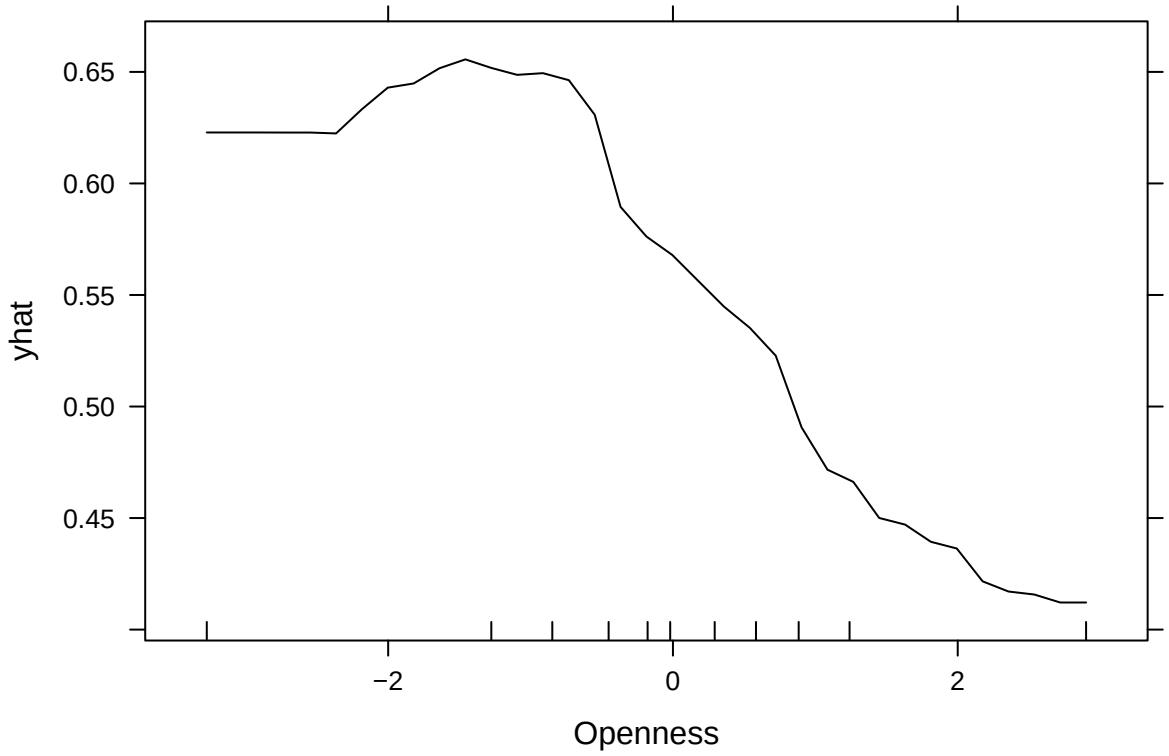

```
pdp4 <- partial(rf, pred.var = "Ethnicity",
                type = "classification",
                which.class = 1, prob = T)
p4 <- plotPartial(pdp4, rug = T, train = drugs_train)
p4
```



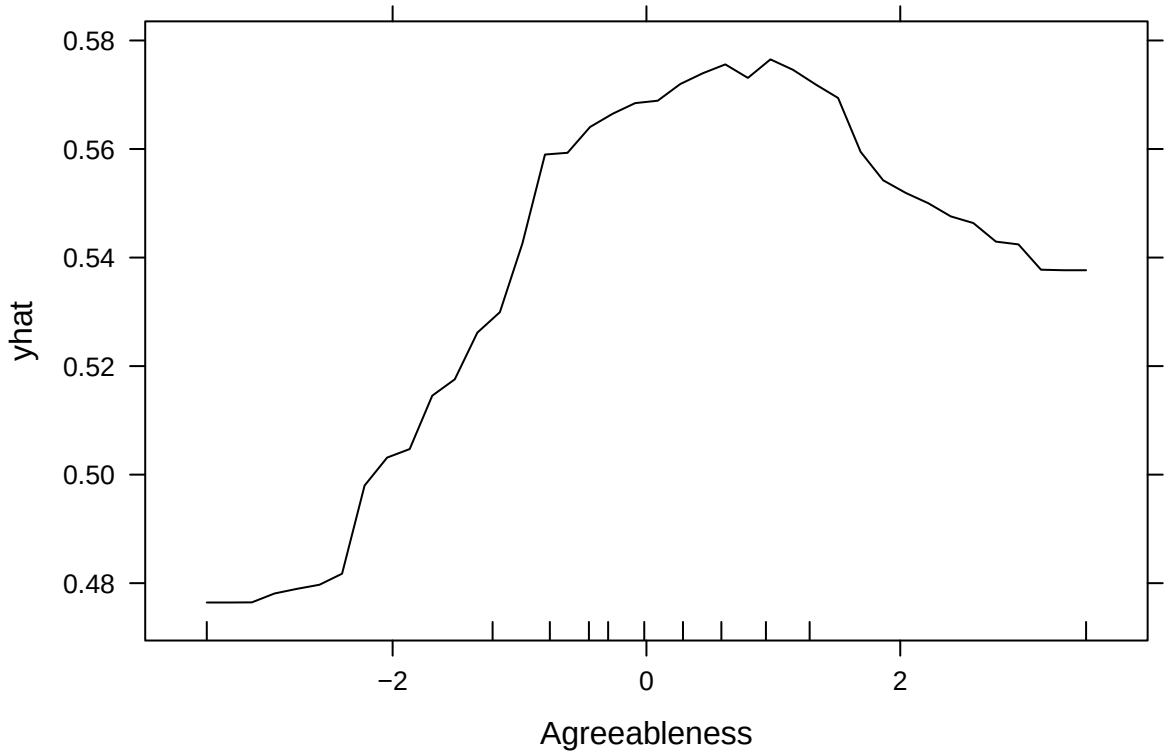

```
pdp5 <- partial(rf, pred.var = c("Neuroticism", "Extraversion"),
                type = "classification",
                which.class = 1, prob = T,
                grid.resolution = 20, progress = "text")
plotPartial(pdp5, levelplot = F, drape = T, colorkey = F,
            screen = list(z = 45, x = -60))
```



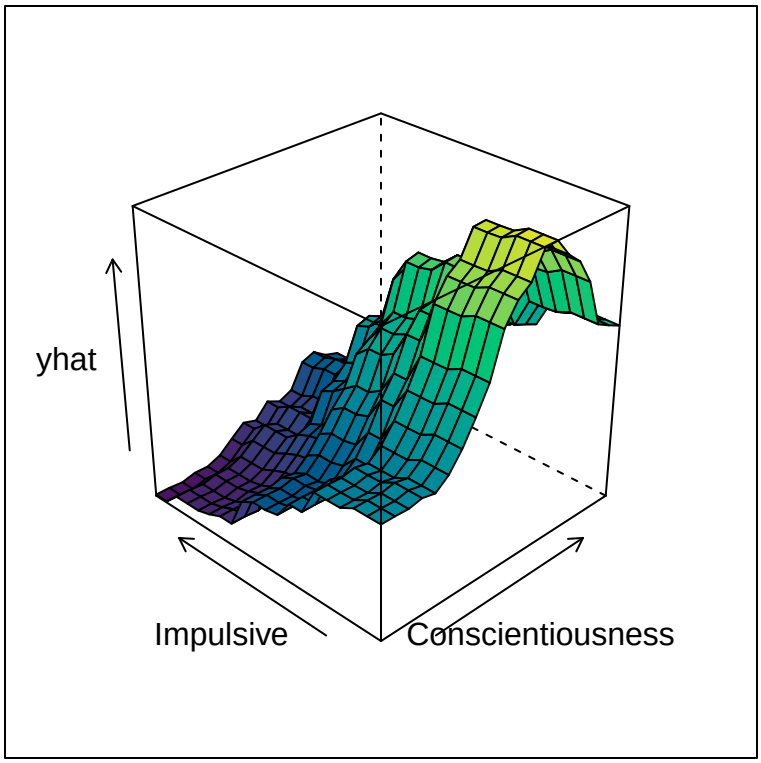

```
pdp6 <- partial(rf, pred.var = c("Openness"),
                type = "classification",
                which.class = 1, prob = T)
p6 <- plotPartial(pdp6, rug = T, train = drugs_train)
p6
```




```
pdp7 <- partial(rf, pred.var = c("Agreeableness"),
                type = "classification",
                which.class = 1, prob = T)
p7 <- plotPartial(pdp7, rug = T, train = drugs_train)
p7
```




```
pdp8 <- partial(rf, pred.var = c("Conscientiousness", "Impulsive"),
                type = "classification",
                which.class = 1, prob = T,
                grid.resolution = 20, progress = "text")
plotPartial(pdp8, levelplot = F, drape = T, colorkey = F,
            screen = list(z = 45, x = -60))
```




```
# The partial dependence plot (short PDP or PD plot) shows the marginal effect one or two features have
```

I focus on plotting the first class, which is CL0. From the plots, female (Gender = 0.48246) seems to have higher probability to have never used LSD than men. People between 45-54 years old have the least probability to have never used LSD, followed by people between 35-44, 55-64, 25-34 in the listed order. Individuals at age 18-24 or 65+ have the highest probabilities to have never used LSD across our sample. People with some college or university, having no certificate or degree, have the least probability to have never used LSD, while people with Master's degree have the highest probability. Black people and Mixed-White/Black people have higher probabilities to have never used LSD across out sample comparing to all other ethnicities. Higher probabilities to have never used LSD are seen at a higher value of 'Extraversion' and a lower value of 'Neuroticism,' which, a positive relationship is observed first, and then followed by a negative relationship. For 'Openness,' a higher value is associated with a lower probability to have never used LSD, while 'Agreeableness' has a reversed result that a higher value is associated with a higher probability to be CL0. Higher 'Conscientiousness' and lower 'Impulsive' are both associated with a higher probability to be CL0.

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

```
pdp9 <- partial(rf, pred.var = "Age",
                type = "classification",
                which.class = 1, prob = T,
                ice = TRUE, center = T)
```

```
## Warning in partial.default(rf, pred.var = "Age", type = "classification", :
## Centering may result in probabilities outside of [0, 1].
```

```
pdp10 <- partial(rf, pred.var = "Gender",
                 type = "classification",
                 which.class = 1, prob = T,
                 ice = TRUE, center = T)
```

```
## Warning in partial.default(rf, pred.var = "Gender", type = "classification", :
## Centering may result in probabilities outside of [0, 1].
```

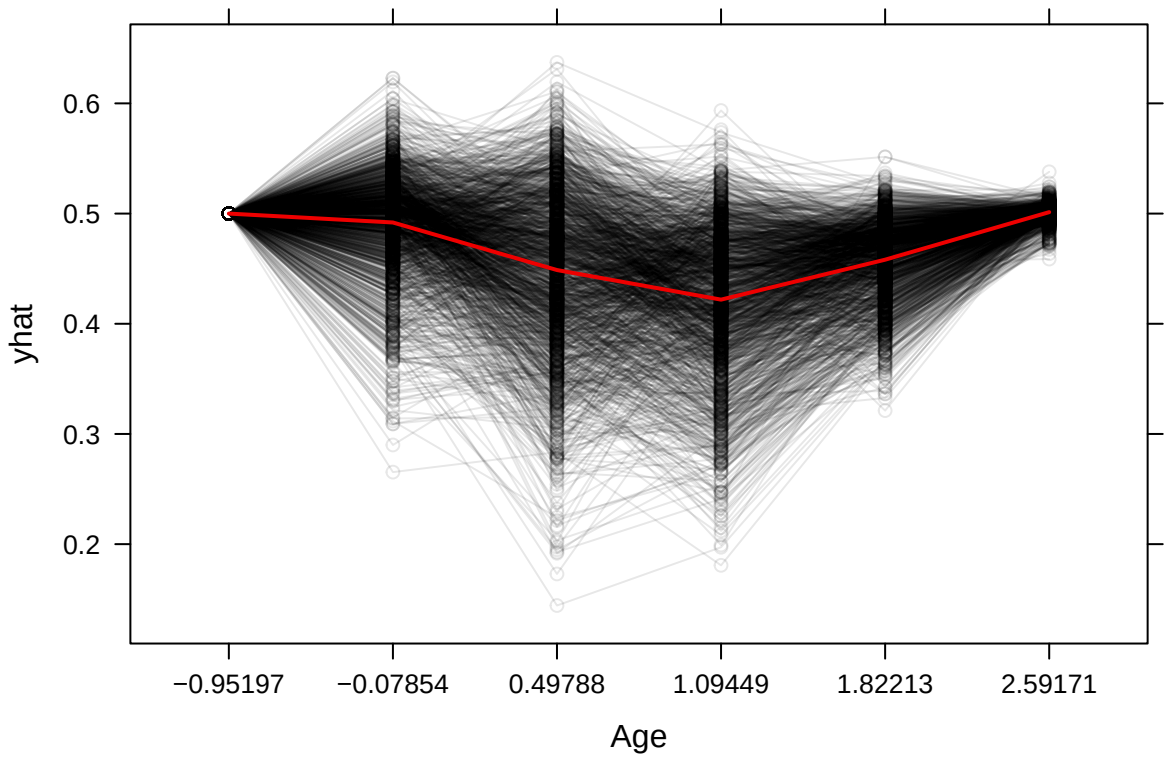
```
pdp11 <- partial(rf, pred.var = "Education",
                 type = "classification",
                 which.class = 1, prob = T,
                 ice = TRUE, center = T)
```

```
## Warning in partial.default(rf, pred.var = "Education", type = "classification",
## : Centering may result in probabilities outside of [0, 1].
```

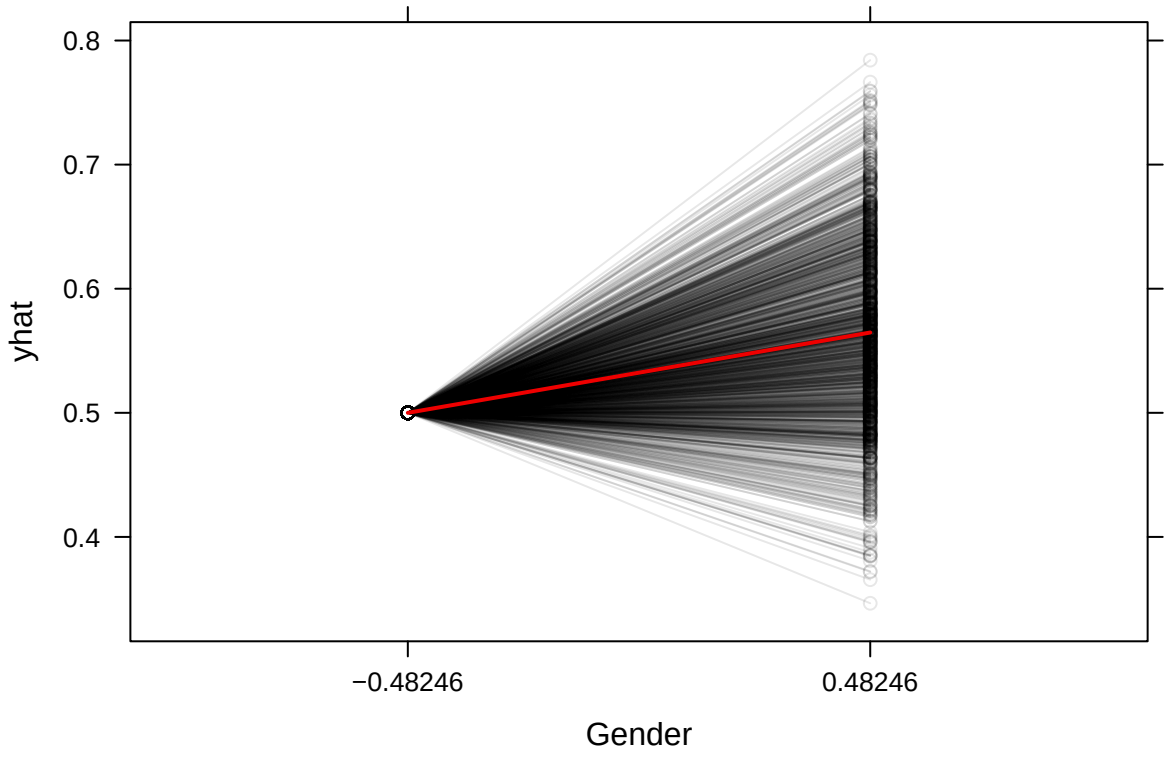
```
pdp12 <- partial(rf, pred.var = "Ethnicity",
                 type = "classification",
                 which.class = 1, prob = T,
                 ice = TRUE, center = T)
```

```
## Warning in partial.default(rf, pred.var = "Ethnicity", type = "classification",
## : Centering may result in probabilities outside of [0, 1].
```

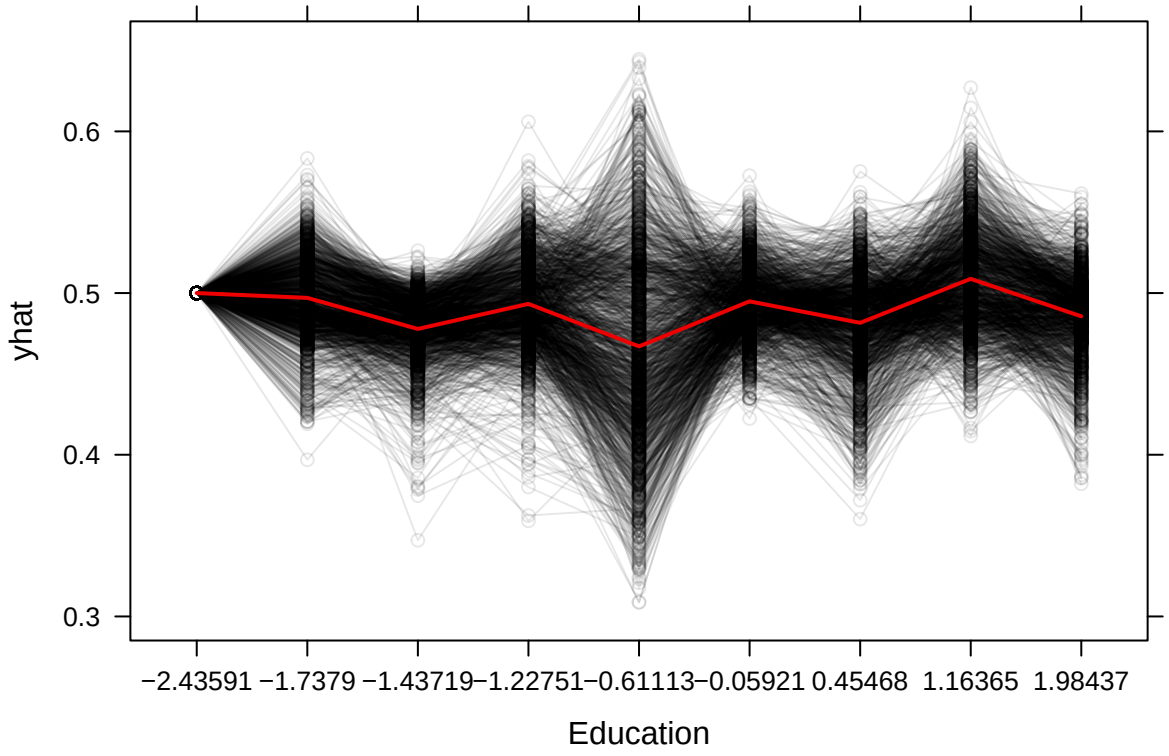
```
plotPartial(pdp9, rug = T, train = drugs_train, alpha = 0.1)
```



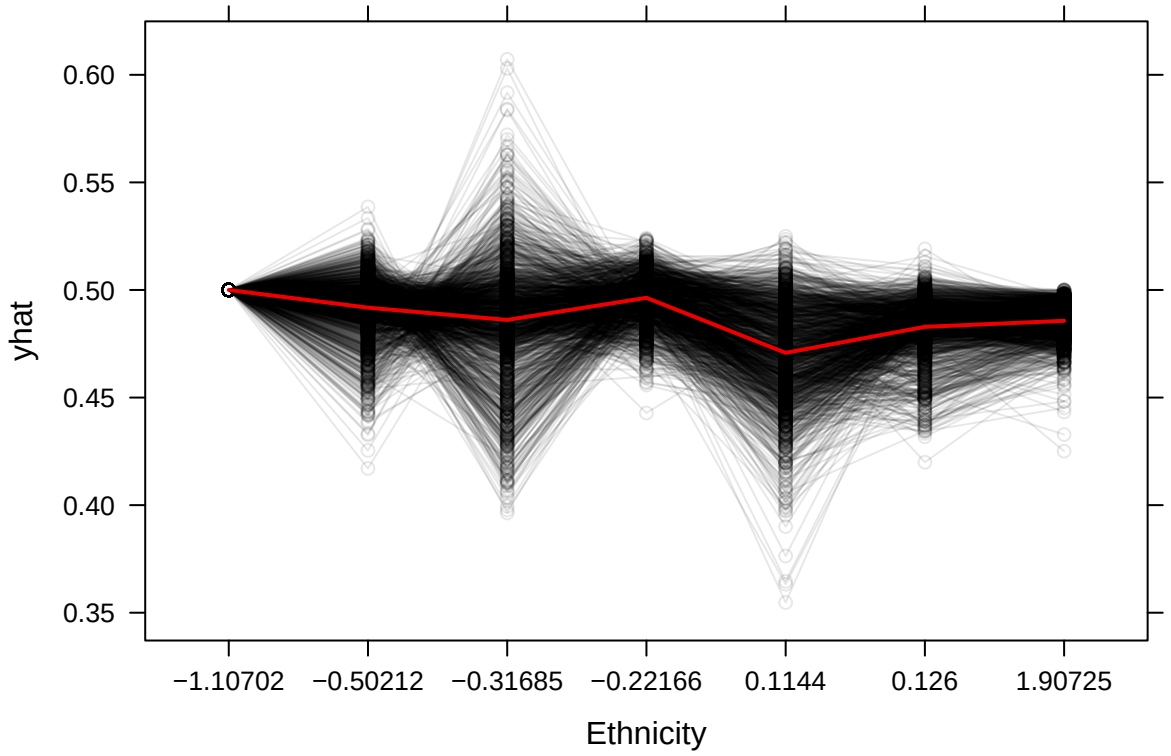
```
plotPartial(pdp10, rug = T, train = drugs_train, alpha = 0.1)
```




```
plotPartial(pdp11, rug = T, train = drugs_train, alpha = 0.1)
```




```
plotPartial(pdp12, rug = T, train = drugs_train, alpha = 0.1)
```



Similar results are seen from the ICE plots comparing to the PDPs, looking at the red line. People between 45-54 years old have the least probability to have never used LSD, while individuals at age 18-24 or 65+ have the highest probabilities to have never used LSD across our sample. In addition, people at age 35-44 or 45-54 have the greatest variances within their groups. Female tends to have a higher probability to have never used LSD than men. People with some college or university, having no certificate or degree, have the least probability to have never used LSD, while people with Master's degree have the highest probability. People who left school at 17 years or have professional certificate/diploma seem to have the least variance within their groups. Black people and Mixed-White/Black people have higher probabilities to have never used LSD across out sample comparing to all other ethnicities, while White people and the 'other' have the greatest variances within their groups.

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

We can remove the least important variables to simplify the model and improve the performance. Interactive terms or transformations may also be included to our model of prediction based on the results of the partial dependence plots. We can also identify the outliers from the ICEs to see what important information we can look at from our sample.

---

3) Prediction and Bias

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

```r
y <- predict(rf, newdata = drugs_test)
y_prob <- predict(rf, newdata = drugs_test, type = "prob")
# y_means <- colMeans(y)
# try fastAdaboost
# ctrl_ada  <- trainControl(method = "cv",
#                           number = 10,
#                           summaryFunction = multiClassSummary,
#                           classProbs = TRUE)
# grid_ada <- expand.grid(nIter = c(50, 100, 150),
#                         method = "Adaboost.M1")
# set.seed(9574)
# ada <- train(LSD ~ Age + Gender + Education + Ethnicity + Country + Neuroticism + Extraversion +
#              Openness + Agreeableness + Conscientiousness + Impulsive + SS,
#              data = drugs_train,
#              method = "adaboost",
#              trControl = ctrl_ada,
#              tuneGrid = grid_ada,
#              metric = "ROC")
# c_ada <- predict(ada, newdata = drugs_test)
# p_ada <- predict(ada, newdata = drugs_test, type = "prob")
```

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

```r
confusionMatrix(y, drugs_test$LSD, mode = "everything", positive = "TRUE")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CL0 CL1 CL2 CL3 CL4 CL5 CL6
##        CL0 201  36  27  30   8   9   1
##        CL1   4  14   1   2   0   0   0
##        CL2   0   1   2   2   0   0   0
##        CL3   8   0   5   8  11   2   1
##        CL4   0   0   0   0   0   0   0
##        CL5   0   0   0   0   0   0   0
##        CL6   0   0   0   0   0   0   0
##
## Overall Statistics
##
##                Accuracy : 0.6032
##                  95% CI : (0.5516, 0.6532)
##     No Information Rate : 0.571
##     P-Value [Acc > NIR] : 0.1142
##
##                   Kappa : 0.2109
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: CL0 Class: CL1 Class: CL2 Class: CL3 Class: CL4
## Sensitivity              0.9437    0.27451   0.057143    0.19048    0.00000
## Specificity              0.3063    0.97826   0.991124    0.91843    1.00000
```

```
## Pos Pred Value           0.6442    0.66667   0.400000    0.22857        NaN
## Neg Pred Value           0.8033    0.89489   0.910326    0.89941    0.94906
## Precision                0.6442    0.66667   0.400000    0.22857         NA
## Recall                   0.9437    0.27451   0.057143    0.19048    0.00000
## F1                       0.7657    0.38889   0.100000    0.20779         NA
## Prevalence               0.5710    0.13673   0.093834    0.11260    0.05094
## Detection Rate           0.5389    0.03753   0.005362    0.02145    0.00000
## Detection Prevalence     0.8365    0.05630   0.013405    0.09383    0.00000
## Balanced Accuracy        0.6250    0.62639   0.524134    0.55445    0.50000
##                      Class: CL5 Class: CL6
## Sensitivity             0.00000   0.000000
## Specificity             1.00000   1.000000
## Pos Pred Value              NaN        NaN
## Neg Pred Value          0.97051   0.994638
## Precision                    NA         NA
## Recall                  0.00000   0.000000
## F1                           NA         NA
## Prevalence              0.02949   0.005362
## Detection Rate          0.00000   0.000000
## Detection Prevalence    0.00000   0.000000
## Balanced Accuracy       0.50000   0.500000
```

The 'confusionMatrix()' contains several evaluation methods for prediction performance, including accuracy score, sensitivity, specificity, precision, and a cross-tabulation of observed and predicted classes. The overall accuracy is 0.6032, which is considered poor in prediction performance.
Since the cost of false positive is high for treating LSD, I will look at the precision. Out of the predicted positives, CL0 and CL1 have higher precision scores (more true positives) than the remaining.
In addition, the cost of false negative is also high for dangering people's life. Our model captures overwhelmingly more actual positives in classes CL0 through labeling them as positive.

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

```
male = drugs_test %>%
        mutate(y = as.factor(y)) %>%
  filter(Gender == -0.48246)

Male <- confusionMatrix(male$y, male$LSD)
Male
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CL0 CL1 CL2 CL3 CL4 CL5 CL6
##        CL0  70  17  15  25   4   8   1
##        CL1   3  12   1   1   0   0   0
##        CL2   0   0   2   2   0   0   0
##        CL3   7   0   5   5  10   2   1
##        CL4   0   0   0   0   0   0   0
##        CL5   0   0   0   0   0   0   0
##        CL6   0   0   0   0   0   0   0
##
## Overall Statistics
```

```
##
##                Accuracy : 0.466
##                  95% CI : (0.3936, 0.5394)
##     No Information Rate : 0.4188
##     P-Value [Acc > NIR] : 0.1066
##
##                   Kappa : 0.1782
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: CL0 Class: CL1 Class: CL2 Class: CL3 Class: CL4
## Sensitivity              0.8750    0.41379    0.08696    0.15152     0.0000
## Specificity              0.3694    0.96914    0.98810    0.84177     1.0000
## Pos Pred Value           0.5000    0.70588    0.50000    0.16667        NaN
## Neg Pred Value           0.8039    0.90230    0.88770    0.82609     0.9267
## Prevalence               0.4188    0.15183    0.12042    0.17277     0.0733
## Detection Rate           0.3665    0.06283    0.01047    0.02618     0.0000
## Detection Prevalence     0.7330    0.08901    0.02094    0.15707     0.0000
## Balanced Accuracy        0.6222    0.69146    0.53753    0.49664     0.5000
##                      Class: CL5 Class: CL6
## Sensitivity             0.00000    0.00000
## Specificity             1.00000    1.00000
## Pos Pred Value              NaN        NaN
## Neg Pred Value          0.94764    0.98953
## Prevalence              0.05236    0.01047
## Detection Rate          0.00000    0.00000
## Detection Prevalence    0.00000    0.00000
## Balanced Accuracy       0.50000    0.50000
```

```
female = drugs_test %>%
        mutate(y = as.factor(y)) %>%
  filter(Gender == 0.48246)

Female <- confusionMatrix(female$y, female$LSD)
Female
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction CL0 CL1 CL2 CL3 CL4 CL5 CL6
##        CL0 131  19  12   5   4   1   0
##        CL1   1   2   0   1   0   0   0
##        CL2   0   1   0   0   0   0   0
##        CL3   1   0   0   3   1   0   0
##        CL4   0   0   0   0   0   0   0
##        CL5   0   0   0   0   0   0   0
##        CL6   0   0   0   0   0   0   0
##
## Overall Statistics
##
##                Accuracy : 0.7473
##                  95% CI : (0.6776, 0.8086)
```

```
##     No Information Rate : 0.7308
##     P-Value [Acc > NIR] : 0.342
##
##                   Kappa : 0.1713
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: CL0 Class: CL1 Class: CL2 Class: CL3 Class: CL4
## Sensitivity              0.9850    0.09091   0.000000    0.33333    0.00000
## Specificity              0.1633    0.98750   0.994118    0.98844    1.00000
## Pos Pred Value           0.7616    0.50000   0.000000    0.60000        NaN
## Neg Pred Value           0.8000    0.88764   0.933702    0.96610    0.97253
## Prevalence               0.7308    0.12088   0.065934    0.04945    0.02747
## Detection Rate           0.7198    0.01099   0.000000    0.01648    0.00000
## Detection Prevalence     0.9451    0.02198   0.005495    0.02747    0.00000
## Balanced Accuracy        0.5741    0.53920   0.497059    0.66089    0.50000
##                      Class: CL5 Class: CL6
## Sensitivity            0.000000         NA
## Specificity            1.000000          1
## Pos Pred Value              NaN         NA
## Neg Pred Value         0.994505         NA
## Prevalence             0.005495          0
## Detection Rate         0.000000          0
## Detection Prevalence   0.000000          0
## Balanced Accuracy      0.500000         NA
```

The two performance metrics do not have similar accuracy score, indicating possible bias in the predictions.