Predicting Recent Nicotine Usage

Ryan Yahnker

Description

The aim of this project is to predict whether an individual recently used nicotine (used within the past year) utilizing supervised learning techniques.

The dataset used includes a total of 1885 observations on 32 variables. A detailed description of the data set can be found below. Each row of the data contains observations of the following demographic and personality trait predictors:

- ID: number of record in original database. Used for reference only.
- Age: Age of the participant
- Gender: Gender of the participant (M/F)
- Education: Level of education of the participant
- Country: Country of current residence of the participant
- Ethnicity: Ethnicity of the participant
- Nscore: NEO-FFI-R Neuroticism (Ranging from 12 to 60)
- Escore: NEO-FFI-R Extraversion (Ranging from 16 to 59)
- Oscore: NEO-FFI-R Openness (Ranging from 24 to 60)
- Ascore: NEO-FFI-R Agreeableness (Ranging from 12 to 60)
- Cscore: NEO-FFI-R Conscientiousness (Ranging from 17 to 59)
- Impulsive: Impulsiveness measured by BIS-11
- SS: Sensation Seeking measured by ImpSS

Participants of the study were also questioned concerning their use of 18 legal and illegal drugs (alcohol, amphetamines, amylnitrite, benzodiazepine, cannabis, chocolate, cocaine, caffeine, crack, ecstasy, heroin, ketamine, legal highs, LSD, methadone, mushrooms, nicotine and volatile substance abuse) and one fictitious drug (Semeron) which was introduced to identify over-claimers. Usage of these drugs were measured on the class system ranging from CL0=CL6 defined below.

- CL0 = "Never Used"
- CL1 = "Used over a decade ago"
- CL2 = "Used in last decade"
- CL3 = "Used in last year"
- CL4 = "Used in last month"
- CL5 = "Used in last week"
- CL6 = "Used in last day".

For this project we will only use the data for nicotine use and train predictive models based on an individuals demographic and personality traits.

Techniques Demonstrated

Data Processing and Feature Engineering, Decision Trees, Boosting Model, Random Forest Model with Bootstrap, Feature Importance Analysis, and Evaluation Metrics (confusion matrix, tpr, fpr)

Set Up

```
#attach necessary packages
suppressPackageStartupMessages({
  library(tidyverse)
  library(glmnet)
  library(tree)
  library(maptree)
  library(randomForest)
  library(gbm)
  library(ROCR)
  library(knitr)
  library(dplyr)})

#set seed for reproducability
set.seed(123)
```

Data Processing and Feature Engineering

```
drugs_data <- read_csv("drug.csv",
                       col_names=c('ID','Age','Gender','Education','Country',
                                   'Ethnicity','Nscore','Escore','Oscore','Ascore',
                                   'Cscore','Impulsive','SS','Alcohol','Amphet',
                                   'Amyl','Benzos','Caff','Cannabis', 'Choc','Coke',
                                   'Crack','Ecstasy','Heroin','Ketamine','Legalh',
                                   'LSD','Meth','Mushrooms','Nicotine','Semer','VSA'))
```

```
## Rows: 1885 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (19): Alcohol, Amphet, Amyl, Benzos, Caff, Cannabis, Choc, Coke, Crack, ...
## dbl (13): ID, Age, Gender, Education, Country, Ethnicity, Nscore, Escore, Os...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(drugs_data)
```

```
## # A tibble: 6 x 32
##      ID     Age Gender Education Country Ethnicity 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
## # i 22 more variables: Cscore <dbl>, Impulsive <dbl>, SS <dbl>, Alcohol <chr>,
## #   Amphet <chr>, Amyl <chr>, Benzos <chr>, Caff <chr>, Cannabis <chr>,
## #   Choc <chr>, Coke <chr>, Crack <chr>, Ecstasy <chr>, Heroin <chr>,
## #   Ketamine <chr>, Legalh <chr>, LSD <chr>, Meth <chr>, Mushrooms <chr>,
## #   Nicotine <chr>, Semer <chr>, VSA <chr>
```

Feature Engineering

```r
#add new factor variable for recent nicotine use utilizing
drugs_data <- drugs_data %>%
  mutate(
    recent_nicotine_use = ifelse(Nicotine >= "CL3", "Yes", "No"),
    recent_nicotine_use = as.factor(recent_nicotine_use)
      )

#check that the variable was added correctly
drugs_data[, c("Nicotine", "recent_nicotine_use")]
```

```
## # A tibble: 1,885 x 2
##    Nicotine recent_nicotine_use
##    <chr>    <fct>
##  1 CL2      No
##  2 CL4      Yes
##  3 CL0      No
##  4 CL2      No
##  5 CL2      No
##  6 CL6      Yes
##  7 CL6      Yes
##  8 CL0      No
##  9 CL6      Yes
## 10 CL6      Yes
## # i 1,875 more rows
```

Data Processing and Splitting

```r
#drugs data subset
drugs_sub <- drugs_data %>%
  dplyr::select(Age:SS, recent_nicotine_use)

#check subset
head(drugs_sub)
```

```
## # A tibble: 6 x 13
##       Age Gender Education Country Ethnicity Nscore Escore  Oscore Ascore
##     <dbl>  <dbl>     <dbl>   <dbl>     <dbl>  <dbl>  <dbl>   <dbl>  <dbl>
## 1  0.498   0.482   -0.0592   0.961     0.126  0.313 -0.575 -0.583  -0.917
## 2 -0.0785 -0.482    1.98     0.961    -0.317 -0.678  1.94   1.44    0.761
## 3  0.498  -0.482   -0.0592   0.961    -0.317 -0.467  0.805 -0.847  -1.62
## 4 -0.952   0.482    1.16     0.961    -0.317 -0.149 -0.806 -0.0193  0.590
## 5  0.498   0.482    1.98     0.961    -0.317  0.735 -1.63  -0.452  -0.302
## 6  2.59    0.482   -1.23     0.249    -0.317 -0.678 -0.300 -1.56    2.04
## # i 4 more variables: Cscore <dbl>, Impulsive <dbl>, SS <dbl>,
## #   recent_nicotine_use <fct>
```

```r
#sample training and testing datasets
drug_train <- sample(nrow(drugs_sub), 1000)
train_data <- drugs_sub[drug_train, ]
test_data <- drugs_sub[-drug_train,]
```

The size of our training dataset is 1000.

The size of our testing dataset is 885.

Decision Tree

```r
#fit a decision tree model
drugs_tree_model <- tree(recent_nicotine_use ~ ., data=train_data)
```

```r
#find best tree size with cross validation
drugs_tree_cv <- cv.tree(drugs_tree_model, FUN=prune.misclass, K=5)
drugs_tree_cv
```

```
## $size
## [1] 8 7 5 2 1
##
## $dev
## [1] 334 334 334 345 431
##
## $k
## [1]  -Inf   0.0   6.5  15.0 107.0
##
## $method
## [1] "misclass"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

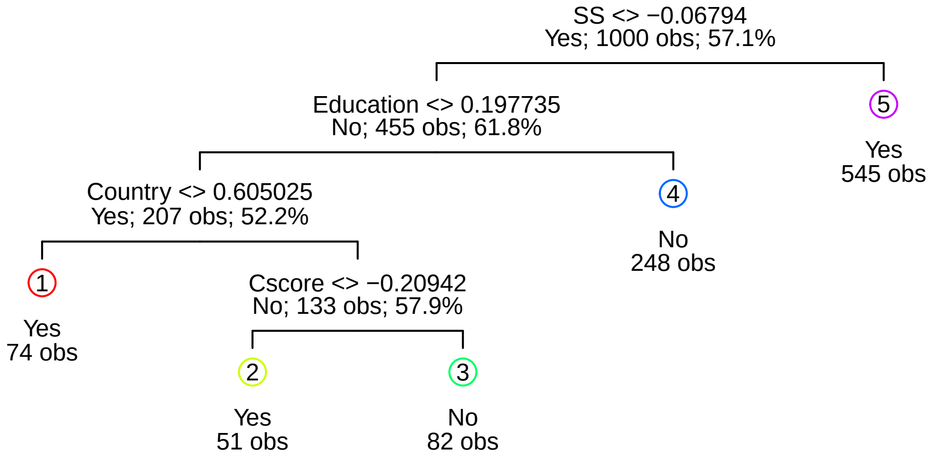
```r
#store minimum CV test error rate and corresponding size
min_error <- min(drugs_tree_cv$dev)
best_size <- min(drugs_tree_cv$size[drugs_tree_cv$dev==min_error])
```

The best tree size obtained through cross validation is 5.

```r
#prune original tree to best size
drugs_tree_model <- prune.misclass(drugs_tree_model, best=best_size)

#plot pruned tree
draw.tree(drugs_tree_model, cex=0.75, nodeinfo=T)
title("Pruned Single Decision Tree", cex.main=0.85)
```

Pruned Single Decision Tree



Evaluation of Decision Tree

```
#confusion matrix
confusion_matrix <- table(Actual=test_data$recent_nicotine_use,
      Predicted=predict(drugs_tree_model, test_data, type="class"))

confusion_matrix
```

```
##       Predicted
## Actual  No Yes
##    No  208 188
##    Yes  96 393
```

```
#calculate FPR TPR
true_pos <- confusion_matrix[2,2]
false_pos <- confusion_matrix[1,2]
true_neg <- confusion_matrix[1,1]
false_neg <- confusion_matrix[2,1]

tpr <- true_pos / (true_pos + false_neg)
fpr <- false_pos / (false_pos + true_neg)

tpr
```

```
## [1] 0.803681
```

```
fpr
```

```
## [1] 0.4747475
```

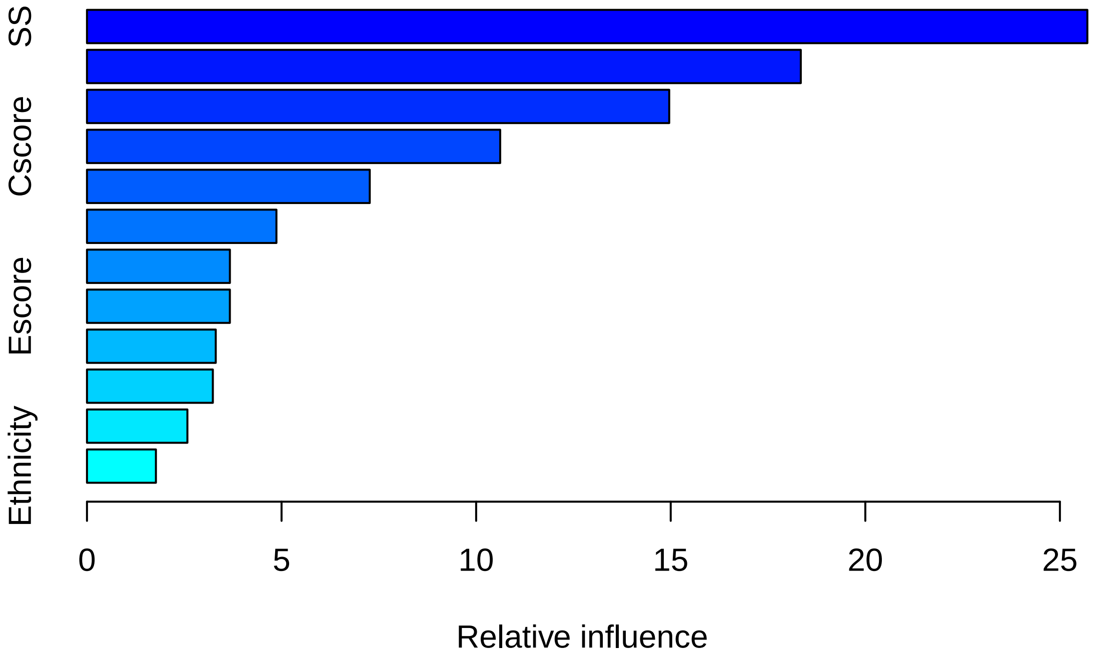
The TPR of the pruned single decision tree is 0.803681.

Boosting Model

```
#assign recent_nicotine_use to numeric {0,1}
train_data$recent_nicotine_use <- as.numeric(train_data$recent_nicotine_use) - 1

#fit boosting model
drugs_boost_model <- gbm(recent_nicotine_use ~ ., data=train_data,
                         distribution="bernoulli", n.trees=1000, shrinkage=0.01)

summary(drugs_boost_model)
```



```
##                        var   rel.inf
## SS                      SS 25.705762
## Country            Country 18.342017
## Age             Age 14.960928
## Cscore       Cscore 10.617483
## Education Education  7.265931
## Oscore       Oscore  4.868171
## Gender       Gender  3.672675
## Escore       Escore  3.671805
## Ascore       Ascore  3.309107
## Impulsive Impulsive  3.234650
## Nscore       Nscore  2.580100
## Ethnicity Ethnicity  1.771369
```

Importance According to Boosting Model

The predictors that appear to have the most importance according to the boosting model are in order SS, Age, Impulsive, and Ascore.

Random Forest Model

```r
#assign recent_nicotine_use to factor
train_data$recent_nicotine_use <- as.factor(train_data$recent_nicotine_use)

#fitting random forest model
drugs_rf_model <- randomForest(recent_nicotine_use ~ ., data=train_data, importance=T)
drugs_rf_model
```

```
##
## Call:
##  randomForest(formula = recent_nicotine_use ~ ., data = train_data,      importance = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 30.1%
## Confusion matrix:
##     0   1 class.error
## 0 252 177   0.4125874
## 1 124 447   0.2171629
```

The out-of-bag estimate of error is `28.2%`. The number of variables randomly considered at each split is `3`. `500` trees were fit into the data.

Importance According to Random Forest Model.

```r
#check variable importance
importance(drugs_rf_model)
```

```
##                    0          1 MeanDecreaseAccuracy MeanDecreaseGini
## Age       15.4672510 15.5189356           22.4539487        38.567248
## Gender    12.5362337  3.4562399           12.3191257        13.769352
## Education  9.0355210  5.9318431           11.6694851        35.353655
## Country   26.7712839  4.3014882           22.3912405        37.643927
## Ethnicity -0.2721216 -0.5264283           -0.5141055         8.747947
## Nscore    -1.1508337  5.1123311            2.8839815        49.835706
## Escore     1.2440362  2.1495490            2.4229249        47.536238
## Oscore     1.7645476  6.9832149            6.6168034        52.430933
## Ascore     5.8955088  0.5144887            4.4330596        49.911096
## Cscore    12.1180044  6.4277455           13.1613557        56.965282
## Impulsive  4.6760172 13.8096694           14.5302435        36.071911
## SS        16.5492237 17.1342396           25.2047729        60.434662
```

SS, Age, Impulsive, and Ascore are still important according to the random forest model, but they are not the most important. The most important predictors differs between the boosting and random forest model.

Evaluation of Boosting model and Random Forest Model

```
#predict probabilities
boost_probabilities <- predict(drugs_boost_model, test_data, type = "response")
```

```
## Using 1000 trees...
```

```
rf_probabilities <- predict(drugs_rf_model, test_data, type="prob")

#use prediction probabilities to assign binary class
boost_class <- ifelse(boost_probabilities >= 0.2, "Yes", "No")
rf_class <- ifelse(rf_probabilities[,2] >= 0.2, "Yes", "No")

#make confusion matrices
boost_cm <- table(Actual=test_data$recent_nicotine_use, Predicted=boost_class)
rf_cm <- table(Actual=test_data$recent_nicotine_use, Predicted=rf_class)

boost_cm
```

```
##       Predicted
## Actual  No Yes
##    No   26 370
##    Yes   8 481
```

```
rf_cm
```

```
##       Predicted
## Actual  No Yes
##    No   44 352
##    Yes  18 471
```

```
#compute TPR for each confusion matrix
boost_tpr <- boost_cm[2, 2] / (boost_cm[2, 2] + boost_cm[2, 1])
rf_tpr <- rf_cm[2, 2] / (rf_cm[2, 2] + rf_cm[2, 1])

boost_tpr
```

```
## [1] 0.9836401
```

```
rf_tpr
```

```
## [1] 0.9631902
```

Conclusion

Our boosting model True Positive Rate is 0.9836401. This value is very high indicating that the boosting model is performing well, correctly predicting “Yes” for recent nicotine usage `98.36`% of the time.

Our random forest model True Positive Rate is 0.9631902. This value is also very high indicating that the random forest model is performing well, correctly predicting “Yes” for recent nicotine usage `96.32`% of the time.

Additionally the boosting and random forest model TPRs are much higher than the TPR of the pruned single decision tree.

Although TPR is an important metric for evaluating model performance, further investigating FPR would be beneficial for better understanding the performance of our models. Fitting an ROC curve and calculating the area under the curve in addition to TPR and FPR would allow us to further interpret the models accuracy when classifying nicotine use. Furthermore, in this project we set our tuning parameter for our boosting model and random forest model to `0.2` which is a lower classification threshold, increasing the models True Positive Rate, but also False Positive Rate. This increased TPR could potentially lead to a false confidence in our models performance, further emphasizing the need to investigate FPR and/or evaluate model performance with larger classification thresholds.

Finally, an important note on the modeling of nicotine use in this project is the failure to use other drug use data. We build our model on soley demographic, personality trait, and nicotine use data of participants. We could potentially build a better model if we build more predictors based off other drug use data. For example if an individual has experimented with many drugs previously, they may be more likely to use nicotine.