BAN502\_Mod4\_RandomForest\_Boersma\_6-7-2021

Jess Boersma

6/7/2021

Dr. Stephen Hill

## Libraries

### Loading the Data

Naming columns

names(drug) = 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")  
  
# str(drug) #(Use shift + Ctrl + C shortcut to comment out)

Replace CL# values to “No” and “Yes”

drug[drug == "CL0"] = "No"  
drug[drug == "CL1"] = "No"  
drug[drug == "CL2"] = "Yes"  
drug[drug == "CL3"] = "Yes"  
drug[drug == "CL4"] = "Yes"  
drug[drug == "CL5"] = "Yes"  
drug[drug == "CL6"] = "Yes"

Factor Conversion and Recoding

* Note the use of **mutate\_at** to target specific ranges of variables.

drug\_clean = drug %>%   
 mutate\_at(vars(Age:Ethnicity), funs(as\_factor)) %>%  
 mutate(Age = factor(Age, labels = c("18\_24", "25\_34", "35\_44", "45\_54", "55\_64", "65\_"))) %>%  
 mutate(Gender = factor(Gender, labels = c("Male", "Female"))) %>%  
 mutate(Education = factor(Education, labels =  
 c("Under16", "At16", "At17", "At18", "SomeCollege",  
 "ProfessionalCert", "Bachelors", "Masters", "Doctorate"))) %>%  
 mutate(Country = factor(Country,  
 labels = c("USA", "NewZealand", "Other", "Australia",  
 "Ireland","Canada","UK"))) %>%  
 mutate(Ethnicity = factor(Ethnicity,  
 labels = c("Black", "Asian", "White", "White/Black", "Other",  
 "White/Asian", "Black/Asian"))) %>%  
 mutate\_at(vars(Alcohol:VSA), funs(as\_factor)) %>%  
 select(-ID)

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

# str(drug\_clean) ## Now only shows variable as factor or numerical.

Removal of drug use variables besides Nicotine use.

drug\_clean = drug\_clean %>%   
 select(!(Alcohol:Mushrooms)) %>%   
 select(!(Semer:VSA))  
names(drug\_clean)

## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Nicotine"

Now down to *13* variables from 32.

## Task 1

Check for missing data in our “drug\_clean” dataframe. Is there any missingness? If so, identify and implement a reasonable strategy to deal with the missingness.

skim for overview.

skim(drug\_clean)

Data summary

|  |  |
| --- | --- |
| Name | drug\_clean |
| Number of rows | 1885 |
| Number of columns | 13 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| factor | 6 |
| numeric | 7 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: factor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| Age | 0 | 1 | FALSE | 6 | 18\_: 643, 25\_: 481, 35\_: 356, 45\_: 294 |
| Gender | 0 | 1 | FALSE | 2 | Mal: 943, Fem: 942 |
| Education | 0 | 1 | FALSE | 9 | Som: 506, Bac: 480, Mas: 283, Pro: 270 |
| Country | 0 | 1 | FALSE | 7 | UK: 1044, USA: 557, Oth: 118, Can: 87 |
| Ethnicity | 0 | 1 | FALSE | 7 | Whi: 1720, Oth: 63, Bla: 33, Asi: 26 |
| Nicotine | 0 | 1 | FALSE | 2 | Yes: 1264, No: 621 |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| Nscore | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.68 | 0.04 | 0.63 | 3.27 | ▁▃▇▅▁ |
| Escore | 0 | 1 | 0.00 | 1.00 | -3.27 | -0.70 | 0.00 | 0.64 | 3.27 | ▁▃▇▃▁ |
| Oscore | 0 | 1 | 0.00 | 1.00 | -3.27 | -0.72 | -0.02 | 0.72 | 2.90 | ▁▃▇▆▁ |
| Ascore | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.61 | -0.02 | 0.76 | 3.46 | ▁▃▇▃▁ |
| Cscore | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.65 | -0.01 | 0.58 | 3.46 | ▁▃▇▃▁ |
| Impulsive | 0 | 1 | 0.01 | 0.95 | -2.56 | -0.71 | -0.22 | 0.53 | 2.90 | ▁▆▇▃▁ |
| SS | 0 | 1 | 0.00 | 0.96 | -2.08 | -0.53 | 0.08 | 0.77 | 1.92 | ▂▇▇▇▅ |

No missingness detected for factors (skim doesn’t catch for Chr) and numerics.

We can also visualize as additional checks. **No missingness noted, so all code commented out**

Simple view of missingess.

# gg\_miss\_var(drug\_clean)

By case:

# gg\_miss\_case(drug\_clean) #x axis is number of missing values in each row (case)

Looking at missingness by variable and combinations of missingness using “aggr” from VIM package.

# vim\_plot = aggr(drug\_clean, numbers = TRUE, prop = c(TRUE, FALSE),cex.axis=.7)  
  
##the cex.axis reduces size of text on x-axis so labels fit better

A view of missingness by variable and row.

# vis\_miss(drug\_clean) #from the naniar package

### 

### Had there been missingness, techniques could include:

* Row-wise deletion
* Column-wise deletion
* Imputation

## 

## Task 2

Split the dataset into training (70%) and testing (30%) sets. Use a set.seed of 1234. Stratify by the “Nicotine” variable.

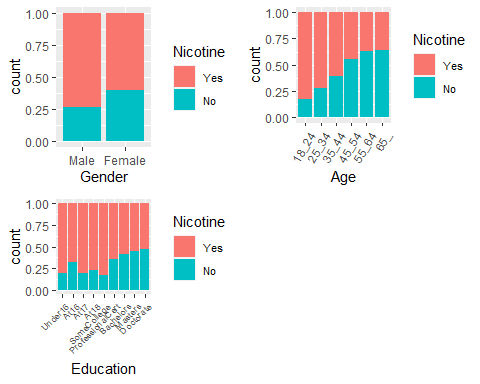
set.seed(1234)  
  
drug\_split <- initial\_split(drug\_clean, prop = 0.7, strata = Nicotine)  
train <- training(drug\_split)  
test <- testing(drug\_split)

## 

## Task 3

Visualize relationships between each variable and “Nicotine” as response.

p1 <- ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle=60, hjust=1)) +  
 theme(legend.text = element\_text(size = 8))   
p2 <- ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill") +  
 theme(legend.text = element\_text(size = 8))   
p3 <- ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle=45, size = 7, hjust=1))+  
 theme(legend.text = element\_text(size = 8))   
  
  
grid.arrange(p2,p1,p3, ncol= 2)



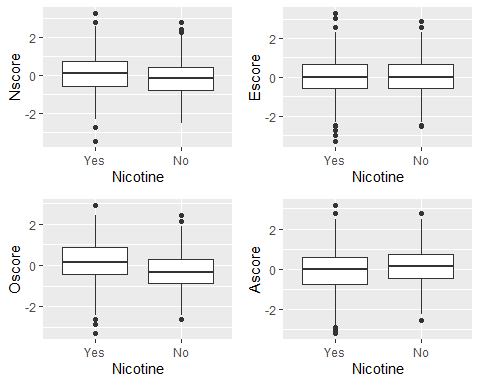
p4 <- ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle=45, hjust=1))  
p5 <-ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(angle=45, hjust=1))  
  
grid.arrange(p4,p5)



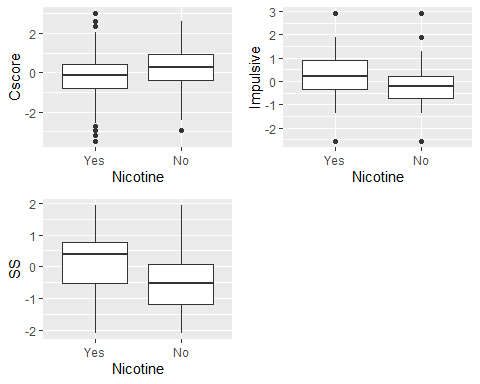
### Commentary 1 of 2: Categorical Variables

All 5 categorical variable visualizations suggest that they are predictors of nicotine use.

p6 = ggplot(train, aes(x = Nicotine, y = Nscore)) + geom\_boxplot()  
p7 = ggplot(train, aes(x = Nicotine, y = Escore)) + geom\_boxplot()  
p8 = ggplot(train, aes(x = Nicotine, y = Oscore)) + geom\_boxplot()  
p9 = ggplot(train, aes(x = Nicotine, y = Ascore)) + geom\_boxplot()  
  
grid.arrange(p6, p7,p8,p9)



p10 = ggplot(train, aes(x = Nicotine, y = Cscore)) + geom\_boxplot()  
p11= ggplot(train, aes(x = Nicotine, y = Impulsive)) + geom\_boxplot()  
p12 = ggplot(train, aes(x = Nicotine, y = SS)) + geom\_boxplot()  
  
  
grid.arrange(p10,p11,p12, ncol = 2)



### Commentary 2 of 2: Numeric Variables

Overall, the numeric variables ranging across the letter+score variables don’t look to be as strong as predictors (with the possible exception of **Cscore**) while the **SS** and **Impulsive** categories do suggest correlation.

That being said, I would not neccessarily rule them out at this point. For example, the Oscore indicates openness to experience and the median for yes is positive and for no is negative. And in the Cscore, which indicates conscientiousness, the median is higher for non-users. This makes common sense within cultural frames of drug use overall.

<http://archive.ics.uci.edu/ml/datasets/Drug+consumption+%28quantified%29>

## Task 4

**Random Forest Model** Create a random forest model on the training set to predict Nicotine using all of the variables in the dataset. You 5-fold, k-fold cross-validation (random number seed of 123 for the folds). Allow R to select mtry values between 2 and 8 and min\_n values between 5 and 20. Use 10 levels in your “grid\_regular” function.

Set a random number seed of 123 for the tune\_grid function. Use 100 trees.

Set up folds (n = 5).

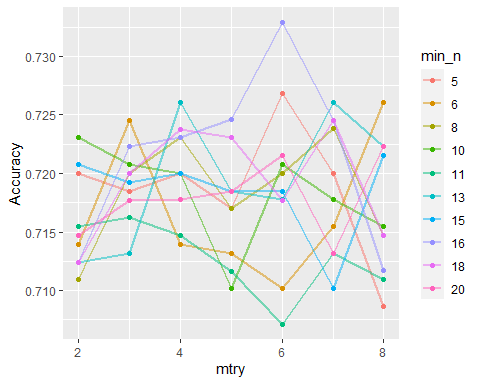
set.seed(123)  
rf\_folds <- vfold\_cv(train, v = 5)

Set Up Model.

drug\_recipe <- recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model <- rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>%  
 set\_engine("ranger", importance = "permutation") %>%  
 set\_mode("classification")  
  
drug\_wflow <-   
 workflow() %>%  
 add\_model(rf\_model) %>%  
 add\_recipe(drug\_recipe)  
  
rf\_grid <- grid\_regular(  
 mtry(range = c(2, 8)),  
 min\_n(range = c(5, 20)),  
 levels = 10  
)  
  
set.seed(123)  
  
rf\_res\_tuned <- tune\_grid(  
 drug\_wflow,  
 resamples = rf\_folds,  
 grid = rf\_grid  
)

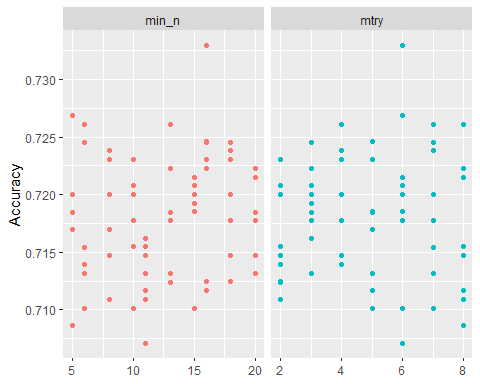
### Visualize the relationships between parameters and performance metrics.

rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 mutate(min\_n = factor(min\_n)) %>%  
 ggplot(aes(mtry, mean, color = min\_n)) +  
 geom\_line(alpha = 0.5, size = 1.05) +  
 geom\_point() +  
 labs(y = "Accuracy")



Best combination appears to be mtry = 6 and min\_n = 16.

rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 select(mean, min\_n, mtry) %>%  
 pivot\_longer(min\_n:mtry,  
 values\_to = "value",  
 names\_to = "parameter"  
 ) %>%  
 ggplot(aes(value, mean, color = parameter)) +  
 geom\_point(show.legend = FALSE) +  
 facet\_wrap(~parameter, scales = "free\_x") +  
 labs(x = NULL, y = "Accuracy")



The graphs are pretty consistent across both parameters with no discernible shape. That being said, doing this in reverse order of the training video by Dr. Hill brings to light how min\_n of 16 and mtry of 6 points are >0.005 points higher than any other accuracy scores.

## Task 5

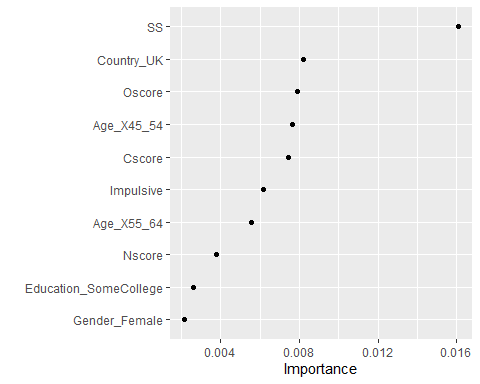
Use the best mtry and min\_n values from Task 4 to finalize the workflow and fit the model to training set. Examine variable importance. What variables are most important in this model?

Select best values to create final random forest for “accuracy”.

best\_rf <- select\_best(rf\_res\_tuned, "accuracy")  
  
# using tune  
final\_rf<- finalize\_workflow(  
 drug\_wflow,  
 best\_rf)  
  
# final\_rf   
## Main arguments match

Fit final\_rf to train data set and examine variable importance.

final\_rf\_fit <- fit(final\_rf, train)  
  
final\_rf\_fit %>% pull\_workflow\_fit() %>% vip(geom = "point")



### Commentary

**Sensation Seeking (SS)** behavior is twice (>0.016) as important as the next variables, with the UK-country factor, openness-to- experience (Oscore), 45-54-age-level factor, conscientiousness (Cscore), Impulsive, and 55-64-age-level variables forming the next visible cluster (~ 0.006-0.008).

## Task 6

How does the model perform on the training and testing sets?

Predictions on Training Dataset, Confusion Matrix, and Comparison with Testing Dataset.

trainpredrf = predict(final\_rf\_fit, train)  
head(trainpredrf)

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 Yes   
## 2 No   
## 3 No   
## 4 No   
## 5 No   
## 6 No

Confusion matrix.

confusionMatrix(trainpredrf$.pred\_class, train$Nicotine,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 870 92  
## No 14 342  
##   
## Accuracy : 0.9196   
## 95% CI : (0.9036, 0.9337)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8092   
##   
## Mcnemar's Test P-Value : 7.495e-14   
##   
## Sensitivity : 0.9842   
## Specificity : 0.7880   
## Pos Pred Value : 0.9044   
## Neg Pred Value : 0.9607   
## Prevalence : 0.6707   
## Detection Rate : 0.6601   
## Detection Prevalence : 0.7299   
## Balanced Accuracy : 0.8861   
##   
## 'Positive' Class : Yes   
##

Predictions on test.

testpredrf = predict(final\_rf\_fit, test)  
head(testpredrf)

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 Yes   
## 3 No   
## 4 No   
## 5 No   
## 6 No

confusionMatrix(testpredrf$.pred\_class, test$Nicotine,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 329 118  
## No 51 69  
##   
## Accuracy : 0.7019   
## 95% CI : (0.6624, 0.7393)  
## No Information Rate : 0.6702   
## P-Value [Acc > NIR] : 0.05808   
##   
## Kappa : 0.2583   
##   
## Mcnemar's Test P-Value : 3.836e-07   
##   
## Sensitivity : 0.8658   
## Specificity : 0.3690   
## Pos Pred Value : 0.7360   
## Neg Pred Value : 0.5750   
## Prevalence : 0.6702   
## Detection Rate : 0.5802   
## Detection Prevalence : 0.7884   
## Balanced Accuracy : 0.6174   
##   
## 'Positive' Class : Yes   
##

### Commentary

The model performs poorly overall. On the training set the accuracy is high at 0.9196 with a ~ 0 p-value. However, the models’ performance on the testing set declines drastically to a score of 0.7019: an accuracy drop of 0.2177 vis-a-vis the training set and a mere 0.0317 points above the no-information rate and a non-significant p-value.

## Task 7

Comment on how this model might be used in the “real-world.” Would you recommend this model for real-world use? What if any concerns would you have about using the model?

### Commentary

I would not recommend using this model at present for numerous reasons:

1. The discrepancy between the training and testing set performances indicates the model is no optimized to work on new data.

* Additional techniques may help address this gap. For example, the number of trees utilized was on the low end. Boehmke & Greenwell (2020) suggest starting with “10 times the number of features” and adjusting from there based on other hyperparameters. <https://bradleyboehmke.github.io/HOML/random-forest.html>
* Other models or mitigation techniques may help with imbalanced dataset.

1. Models with real-world implications involving human beings engender a whole series of ethical considerations. I have the following concerns:

* Qualified health professionals should be involved in providing input and oversight regarding the validity of the model’s findings and implementation. Just as database specialists don’t create the logic models, nor should data analysts be at the helm of the implementation of mode.
* As an illustrative example, the strongest predictor, “SS” is based on the ImpSS scale <http://archive.ics.uci.edu/ml/datasets/Drug+consumption+%28quantified%29>, yet its validity has not been defintively established as being superior to other methods such as the Sensation Seeking Scale, Form V (SSS-V). Articles (e.g., McDaniel & Mahan 2008) do suggest comparability, but again, that’s the realm of properly trained experts in psychometrics and psychobiology.