# David Wells

## Module 4 Assignment 4

Load-in Libraries

Load-in Data and Cleaning

drug <- read\_csv("~/BAN 502/Module 4/drug\_data-1.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## .default = col\_character(),  
## Column1 = col\_double(),  
## Column2 = col\_double(),  
## Column3 = col\_double(),  
## Column4 = col\_double(),  
## Column5 = col\_double(),  
## Column6 = col\_double(),  
## Column7 = col\_double(),  
## Column8 = col\_double(),  
## Column9 = col\_double(),  
## Column10 = col\_double(),  
## Column11 = col\_double(),  
## Column12 = col\_double(),  
## Column13 = col\_double()  
## )  
## i Use `spec()` for the full column specifications.

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")  
  
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"  
  
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()` is deprecated as of 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))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

#str(drug\_clean)  
  
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"

Task 1

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 | ▂▇▇▇▅ |

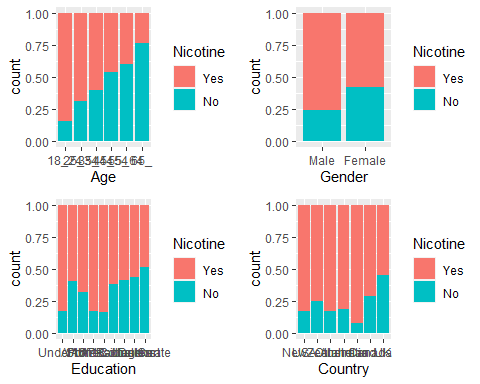
# There doesn't seem to be any missing data.

Task 2

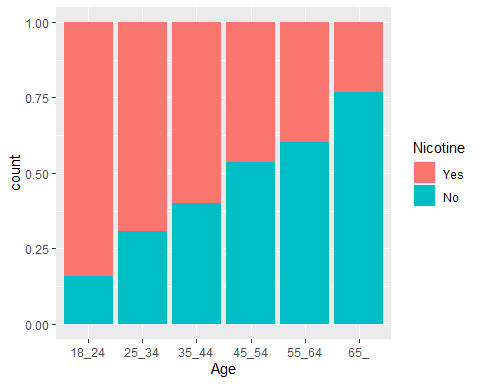
set.seed(1234)   
drugc\_split = initial\_split(drug\_clean, prop = 0.7, strata = Nicotine) #70% in training  
train = training(drugc\_split)  
test = testing(drugc\_split)

Task 3

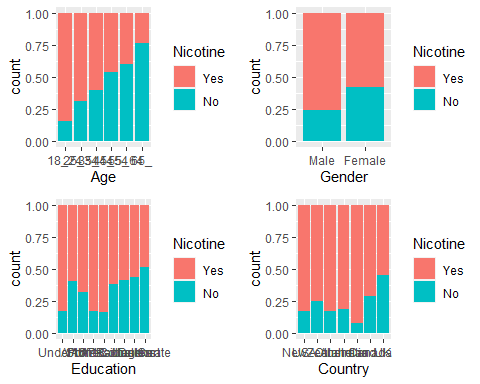
p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1,p2,p3,p4)



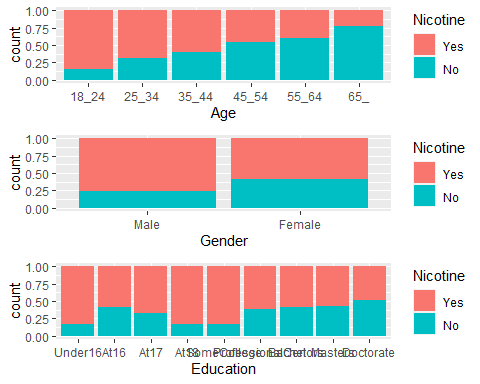
p5 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1)



p6 = ggplot(train, aes(x = Nscore, fill = Nicotine)) + geom\_bar(position = "fill")  
p7 = ggplot(train, aes(x = Escore, fill = Nicotine)) + geom\_bar(position = "fill")  
p8 = ggplot(train, aes(x = Oscore, fill = Nicotine)) + geom\_bar(position = "fill")  
p9 = ggplot(train, aes(x = Ascore, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1,p2,p3,p4)



p10 = ggplot(train, aes(x = Cscore, fill = Nicotine)) + geom\_bar(position = "fill")  
p11 = ggplot(train, aes(x = Impulsive, fill = Nicotine)) + geom\_bar(position = "fill")  
p12 = ggplot(train, aes(x = SS, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1,p2,p3)



# From the visualizations, Age, Gender, Education, Country, and Ethnicity all seem to have some relationship with Nicotine use. Looking at the rest of the variables, it only looks like Escore does not have a relationship with Nicotine use.

Task 4

set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)  
  
drug\_recipe = recipe(Nicotine ~., train)  
  
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   
)

##   
## Attaching package: 'rlang'

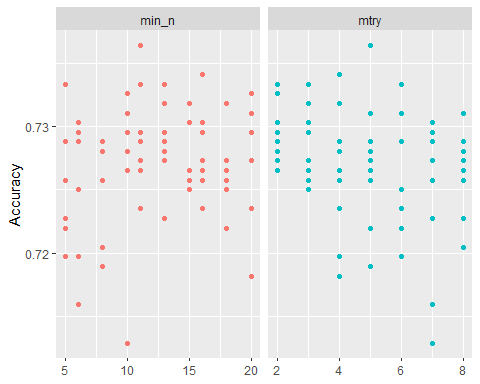
## The following objects are masked from 'package:purrr':  
##   
## %@%, as\_function, flatten, flatten\_chr, flatten\_dbl, flatten\_int,  
## flatten\_lgl, flatten\_raw, invoke, list\_along, modify, prepend,  
## splice

##   
## Attaching package: 'vctrs'

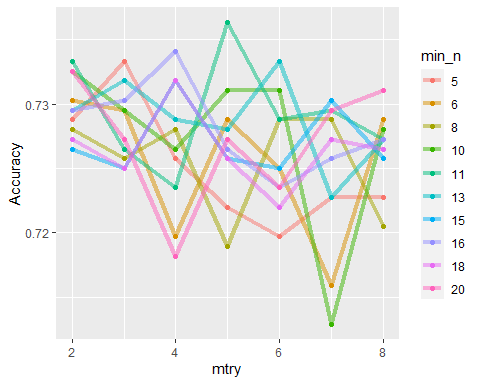
## The following object is masked from 'package:dplyr':  
##   
## data\_frame

## The following object is masked from 'package:tibble':  
##   
## data\_frame

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")



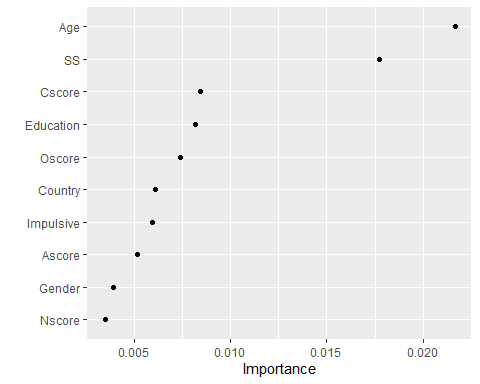
rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 mutate(min\_n = factor(min\_n)) %>%  
 ggplot(aes(mtry, mean, color = min\_n)) +  
 geom\_line(alpha = 0.5, size = 1.5) +  
 geom\_point() +  
 labs(y = "Accuracy")

 # Task 5

best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 drug\_wflow,  
 best\_rf  
)  
  
final\_rf

## == Workflow ====================================================================  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## -- Preprocessor ----------------------------------------------------------------  
## 0 Recipe Steps  
##   
## -- Model -----------------------------------------------------------------------  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 5  
## trees = 100  
## min\_n = 11  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

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



# From the chart, it appears that Age, Sensation Seeing (SS), and Openess to Experience (Oscore) are the most important variables.

Task 6

#Predictions on Training Set  
trainpredrf = predict(final\_rf\_fit, train)  
head(trainpredrf)

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 881 35  
## No 4 400  
##   
## Accuracy : 0.9705   
## 95% CI : (0.9598, 0.9789)  
## No Information Rate : 0.6705   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9319   
##   
## Mcnemar's Test P-Value : 1.556e-06   
##   
## Sensitivity : 0.9955   
## Specificity : 0.9195   
## Pos Pred Value : 0.9618   
## Neg Pred Value : 0.9901   
## Prevalence : 0.6705   
## Detection Rate : 0.6674   
## Detection Prevalence : 0.6939   
## Balanced Accuracy : 0.9575   
##   
## 'Positive' Class : Yes   
##

# Predictions on Testing Set  
testpredrf = predict(final\_rf\_fit, test)  
head(testpredrf)

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 322 97  
## No 57 89  
##   
## Accuracy : 0.7274   
## 95% CI : (0.6887, 0.7638)  
## No Information Rate : 0.6708   
## P-Value [Acc > NIR] : 0.002117   
##   
## Kappa : 0.3471   
##   
## Mcnemar's Test P-Value : 0.001674   
##   
## Sensitivity : 0.8496   
## Specificity : 0.4785   
## Pos Pred Value : 0.7685   
## Neg Pred Value : 0.6096   
## Prevalence : 0.6708   
## Detection Rate : 0.5699   
## Detection Prevalence : 0.7416   
## Balanced Accuracy : 0.6640   
##   
## 'Positive' Class : Yes   
##

# The model had a 97% accuracy on the Training Set. The model had a 71% accuracy on the Testing Set. Assuming would only yield a 67% accuracy rating. I don't feel very confident in this model.

Task 7

# The model as is could be used to determine nicotine use in health insurance calculations, for example. The model may be able to accurately identify those individuals who do use nicotine; it may also however classify non-nicotine users as nicotine users. It may be, from a business case, better to over-classify nicotine users. Those individuals who are not nicotine users could have the classification changed with a minimum of impact.   
  
# I would not use this model, as I don't think it is precise enough. I would rather, and this is entirely a gut-feeling, have the Testing Set accuracy percentage in the 80+ range.