# Module 4 Assignment 2

## Underwood, Katie

drug = read\_csv("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")  
  
#str(drug)  
  
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

No missingness in the data

#summary(drug\_clean)  
#library(skimr)  
#skim(drug\_clean)

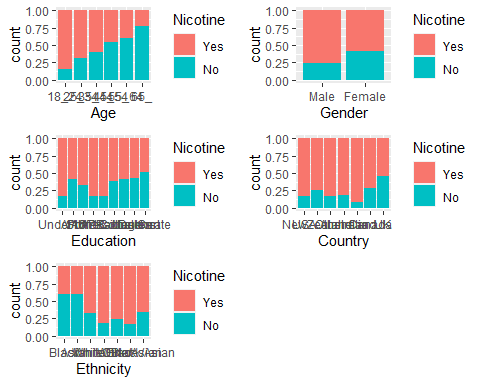
## Task 2

set.seed(1234)  
drug\_split = initial\_split(drug\_clean, prop = .7, strata = Nicotine)  
train = training(drug\_split)  
test = testing(drug\_split)

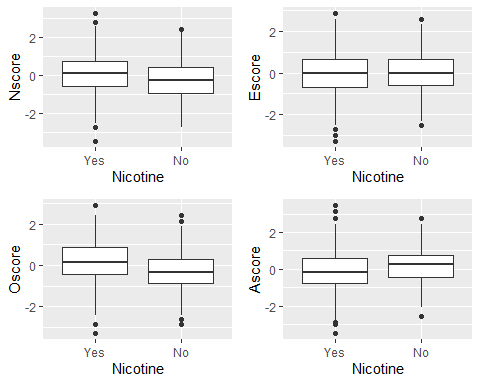
## Task 3

Age, Gender, Education, Country, Ethnicity all seem to have an effect on the rate of Nicotine use.

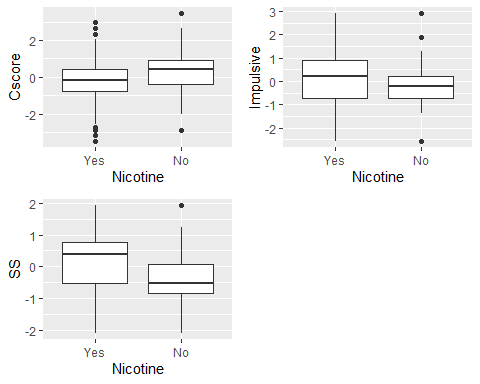
p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill")  
p5 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1,p2,p3,p4,p5)

 Nicotine users have slightly higher Nscore and Oscore, maybe a slightly lower Ascore but not very significant.

p1 = ggplot(train, aes(x = Nicotine, y=Nscore)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y=Escore)) + geom\_boxplot()  
p3 = ggplot(train, aes(x = Nicotine, y=Oscore)) + geom\_boxplot()  
p4 = ggplot(train, aes(x = Nicotine, y=Ascore)) + geom\_boxplot()  
grid.arrange(p1,p2,p3,p4)

 Impulsive and SS have interesting differences between Nicotine use/non-use

p1 = ggplot(train, aes(x = Nicotine, y=Cscore)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y=Impulsive)) + geom\_boxplot()  
p3 = ggplot(train, aes(x = Nicotine, y=SS)) + geom\_boxplot()  
grid.arrange(p1,p2,p3, ncol = 2)



## Task 4

set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)  
  
nicotine\_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")  
  
nicotine\_wfow =  
 workflow() %>%  
 add\_model(rf\_model) %>%  
 add\_recipe(nicotine\_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(  
 nicotine\_wfow,   
 resamples = rf\_folds,   
 grid = rf\_grid)

##   
## Attaching package: 'rlang'

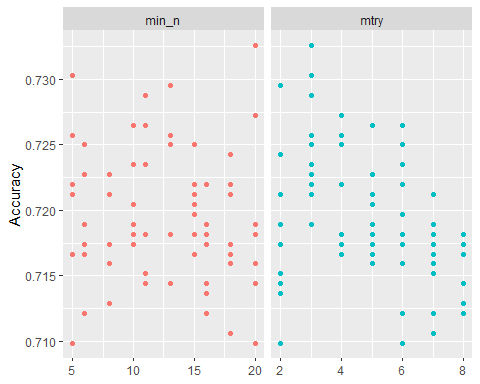
## The following objects are masked from 'package:purrr':  
##   
## %@%, as\_function, flatten, flatten\_chr, flatten\_dbl, flatten\_int,  
## flatten\_lgl, flatten\_raw, invoke, list\_along, modify, prepend,  
## splice

##   
## Attaching package: 'vctrs'

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

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

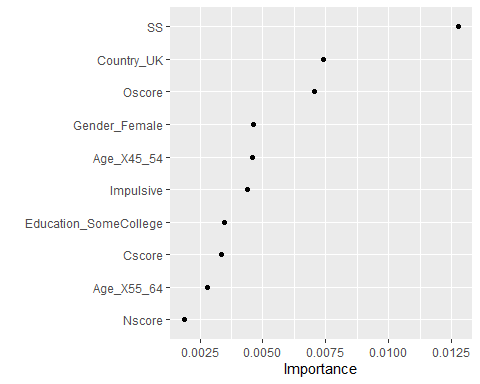
rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 select(mean, min\_n, mtry) %>%  
 pivot\_longer(min\_n:mtry,  
 values\_to = "value",  
 names\_to = "parameter"  
 ) %>%  
 ggplot(aes(value, mean, color = parameter)) +  
 geom\_point(show.legend = FALSE) +  
 facet\_wrap(~parameter,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



## Task 5

SS (sensation seeing) seems to be the most important variable by a decent margin, followed by county = UK or not, followed by Oscore and age.

best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 nicotine\_wfow,  
 best\_rf  
)  
  
final\_rf\_fit = fit(final\_rf, train)  
  
final\_rf\_fit %>% pull\_workflow\_fit() %>% vip(geom = "point")



## Task 6

Model has accuracy of .84 on training set and .71 on the testing set.

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

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 850 171  
## No 35 264  
##   
## Accuracy : 0.8439   
## 95% CI : (0.8232, 0.8631)  
## No Information Rate : 0.6705   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6163   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9605   
## Specificity : 0.6069   
## Pos Pred Value : 0.8325   
## Neg Pred Value : 0.8829   
## Prevalence : 0.6705   
## Detection Rate : 0.6439   
## Detection Prevalence : 0.7735   
## Balanced Accuracy : 0.7837   
##   
## 'Positive' Class : Yes   
##

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

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 Yes   
## 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 337 121  
## No 42 65  
##   
## Accuracy : 0.7115   
## 95% CI : (0.6722, 0.7486)  
## No Information Rate : 0.6708   
## P-Value [Acc > NIR] : 0.02111   
##   
## Kappa : 0.2676   
##   
## Mcnemar's Test P-Value : 9.999e-10   
##   
## Sensitivity : 0.8892   
## Specificity : 0.3495   
## Pos Pred Value : 0.7358   
## Neg Pred Value : 0.6075   
## Prevalence : 0.6708   
## Detection Rate : 0.5965   
## Detection Prevalence : 0.8106   
## Balanced Accuracy : 0.6193   
##   
## 'Positive' Class : Yes   
##

## Task 7

The accuracy of this model is above the naive rate, so it might be OK to use in the real world. One good thing about the model is the high sensitivity, there will be only a few smokers not detected by this model This would be troublesome for an application like smoking prevention. A potential problem I see is the specificity, at only .59 there will be a lot of “false positives” which means many non-smokers will be identified as smokers. This would be troublesome for an application like identifying life insurance premiums, the non-smokers would get the higher smoking rate for insurance.