#install.packages("gridExtra")  
#install.packages("vip")  
#install.packages("ranger")

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 0.1.2 ──

## ✓ broom 0.7.2 ✓ recipes 0.1.15  
## ✓ dials 0.0.9 ✓ rsample 0.0.8   
## ✓ infer 0.5.4 ✓ tune 0.1.2   
## ✓ modeldata 0.1.0 ✓ workflows 0.2.1   
## ✓ parsnip 0.1.5 ✓ yardstick 0.0.7

## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity

## The following object is masked from 'package:purrr':  
##   
## lift

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(vip)

##   
## Attaching package: 'vip'

## The following object is masked from 'package:utils':  
##   
## vi

library(ranger)

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)

## tibble [1,885 × 32] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ID : num [1:1885] 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : num [1:1885] 0.4979 -0.0785 0.4979 -0.952 0.4979 ...  
## $ Gender : num [1:1885] 0.482 -0.482 -0.482 0.482 0.482 ...  
## $ Education: num [1:1885] -0.0592 1.9844 -0.0592 1.1637 1.9844 ...  
## $ Country : num [1:1885] 0.961 0.961 0.961 0.961 0.961 ...  
## $ Ethnicity: num [1:1885] 0.126 -0.317 -0.317 -0.317 -0.317 ...  
## $ Nscore : num [1:1885] 0.313 -0.678 -0.467 -0.149 0.735 ...  
## $ Escore : num [1:1885] -0.575 1.939 0.805 -0.806 -1.633 ...  
## $ Oscore : num [1:1885] -0.5833 1.4353 -0.8473 -0.0193 -0.4517 ...  
## $ Ascore : num [1:1885] -0.917 0.761 -1.621 0.59 -0.302 ...  
## $ Cscore : num [1:1885] -0.00665 -0.14277 -1.0145 0.58489 1.30612 ...  
## $ Impulsive: num [1:1885] -0.217 -0.711 -1.38 -1.38 -0.217 ...  
## $ SS : num [1:1885] -1.181 -0.216 0.401 -1.181 -0.216 ...  
## $ Alcohol : chr [1:1885] "CL5" "CL5" "CL6" "CL4" ...  
## $ Amphet : chr [1:1885] "CL2" "CL2" "CL0" "CL0" ...  
## $ Amyl : chr [1:1885] "CL0" "CL2" "CL0" "CL0" ...  
## $ Benzos : chr [1:1885] "CL2" "CL0" "CL0" "CL3" ...  
## $ Caff : chr [1:1885] "CL6" "CL6" "CL6" "CL5" ...  
## $ Cannabis : chr [1:1885] "CL0" "CL4" "CL3" "CL2" ...  
## $ Choc : chr [1:1885] "CL5" "CL6" "CL4" "CL4" ...  
## $ Coke : chr [1:1885] "CL0" "CL3" "CL0" "CL2" ...  
## $ Crack : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ Ecstasy : chr [1:1885] "CL0" "CL4" "CL0" "CL0" ...  
## $ Heroin : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ Ketamine : chr [1:1885] "CL0" "CL2" "CL0" "CL2" ...  
## $ Legalh : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ LSD : chr [1:1885] "CL0" "CL2" "CL0" "CL0" ...  
## $ Meth : chr [1:1885] "CL0" "CL3" "CL0" "CL0" ...  
## $ Mushrooms: chr [1:1885] "CL0" "CL0" "CL1" "CL0" ...  
## $ Nicotine : chr [1:1885] "CL2" "CL4" "CL0" "CL2" ...  
## $ Semer : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## $ VSA : chr [1:1885] "CL0" "CL0" "CL0" "CL0" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. 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(),  
## .. Column14 = col\_character(),  
## .. Column15 = col\_character(),  
## .. Column16 = col\_character(),  
## .. Column17 = col\_character(),  
## .. Column18 = col\_character(),  
## .. Column19 = col\_character(),  
## .. Column20 = col\_character(),  
## .. Column21 = col\_character(),  
## .. Column22 = col\_character(),  
## .. Column23 = col\_character(),  
## .. Column24 = col\_character(),  
## .. Column25 = col\_character(),  
## .. Column26 = col\_character(),  
## .. Column27 = col\_character(),  
## .. Column28 = col\_character(),  
## .. Column29 = col\_character(),  
## .. Column30 = col\_character(),  
## .. Column31 = col\_character(),  
## .. Column32 = col\_character()  
## .. )

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)

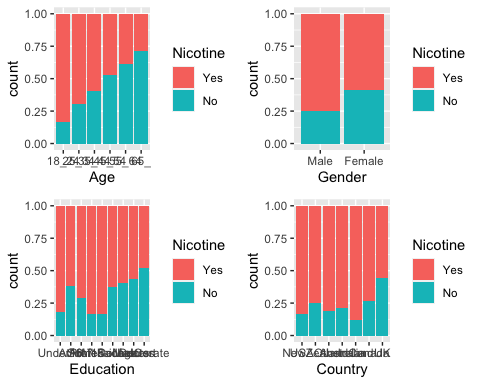
## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Nicotine"

summary(drug\_clean)

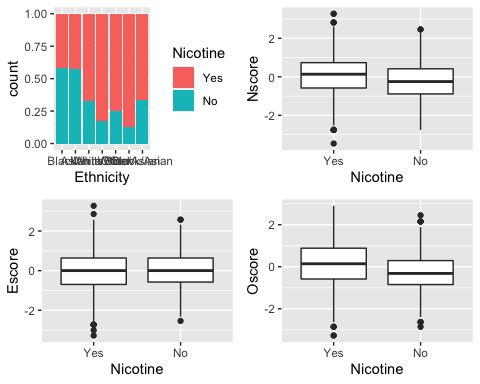
## Age Gender Education Country   
## 18\_24:643 Male :943 SomeCollege :506 USA : 557   
## 25\_34:481 Female:942 Bachelors :480 NewZealand: 5   
## 35\_44:356 Masters :283 Other : 118   
## 45\_54:294 ProfessionalCert:270 Australia : 54   
## 55\_64: 93 At18 :100 Ireland : 20   
## 65\_ : 18 At16 : 99 Canada : 87   
## (Other) :147 UK :1044   
## Ethnicity Nscore Escore Oscore   
## Black : 33 Min. :-3.464360 Min. :-3.273930 Min. :-3.273930   
## Asian : 26 1st Qu.:-0.678250 1st Qu.:-0.695090 1st Qu.:-0.717270   
## White :1720 Median : 0.042570 Median : 0.003320 Median :-0.019280   
## White/Black: 20 Mean : 0.000047 Mean :-0.000163 Mean :-0.000534   
## Other : 63 3rd Qu.: 0.629670 3rd Qu.: 0.637790 3rd Qu.: 0.723300   
## White/Asian: 20 Max. : 3.273930 Max. : 3.273930 Max. : 2.901610   
## Black/Asian: 3   
## Ascore Cscore Impulsive   
## Min. :-3.464360 Min. :-3.464360 Min. :-2.555240   
## 1st Qu.:-0.606330 1st Qu.:-0.652530 1st Qu.:-0.711260   
## Median :-0.017290 Median :-0.006650 Median :-0.217120   
## Mean :-0.000245 Mean :-0.000386 Mean : 0.007216   
## 3rd Qu.: 0.760960 3rd Qu.: 0.584890 3rd Qu.: 0.529750   
## Max. : 3.464360 Max. : 3.464360 Max. : 2.901610   
##   
## SS Nicotine   
## Min. :-2.078480 Yes:1264   
## 1st Qu.:-0.525930 No : 621   
## Median : 0.079870   
## Mean :-0.003292   
## 3rd Qu.: 0.765400   
## Max. : 1.921730   
##

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

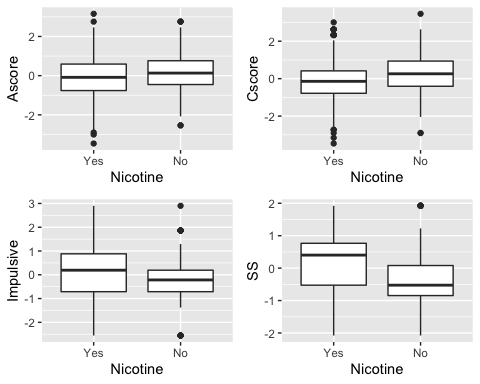
p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill =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



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



p1 = ggplot(train, aes(x = Nicotine, y = Ascore)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y = Cscore)) + geom\_boxplot()  
p3 = ggplot(train, aes(x = Nicotine, y = Impulsive)) + geom\_boxplot()  
p4 = ggplot(train, aes(x = Nicotine, y = SS)) + geom\_boxplot()  
grid.arrange(p1,p2,p3,p4)



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

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)  
  
set.seed(123)  
rf\_res = tune\_grid(  
 drug\_wflow,  
 resamples = rf\_folds,  
 grid = 20  
)

## i Creating pre-processing data to finalize unknown parameter: mtry

##   
## Attaching package: 'rlang'

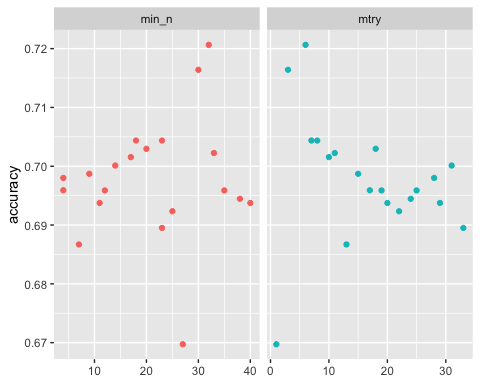
## The following objects are masked from 'package:purrr':  
##   
## %@%, as\_function, flatten, flatten\_chr, flatten\_dbl, flatten\_int,  
## flatten\_lgl, flatten\_raw, invoke, list\_along, modify, prepend,  
## splice

##   
## Attaching package: 'vctrs'

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

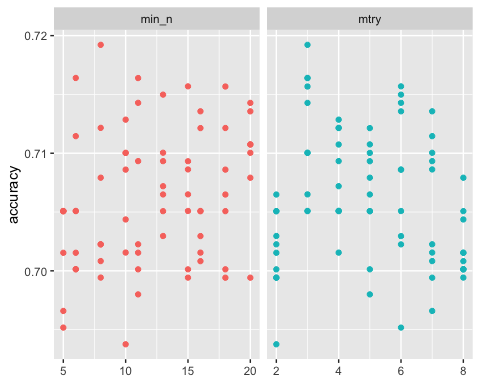
## The following object is masked from 'package:tibble':  
##   
## data\_frame

rf\_res %>%  
 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")



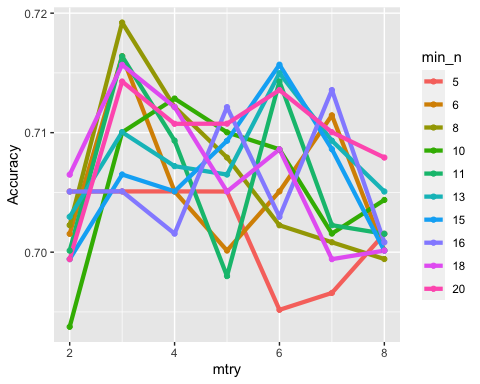
drug\_recipe = recipe(Nicotine ~ ., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>%  
 set\_engine("ranger", importance = "permutation") %>%  
 set\_mode("classification")  
  
drug\_wflow =   
 workflow() %>%  
 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  
)

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(aplha = .5, size = 1.5) +  
 geom\_point() +  
 labs(y = "Accuracy")

## Warning: Ignoring unknown parameters: aplha

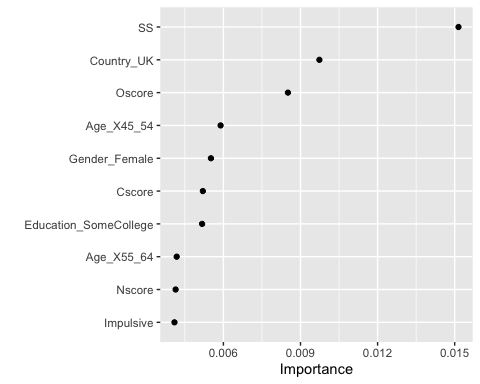


best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 drug\_wflow,  
 best\_rf  
)  
  
final\_rf

## ══ Workflow ════════════════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 1 Recipe Step  
##   
## ● step\_dummy()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 3  
## trees = 100  
## min\_n = 8  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

final\_rf\_fit = fit(final\_rf, train)

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



The most important variables are SS, Country\_UK, and OScore.

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

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 Yes   
## 2 Yes   
## 3 Yes   
## 4 Yes   
## 5 Yes   
## 6 Yes

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 933 103  
## No 15 363  
##   
## Accuracy : 0.9165   
## 95% CI : (0.9009, 0.9304)  
## No Information Rate : 0.6704   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8016   
##   
## Mcnemar's Test P-Value : 1.156e-15   
##   
## Sensitivity : 0.9842   
## Specificity : 0.7790   
## Pos Pred Value : 0.9006   
## Neg Pred Value : 0.9603   
## Prevalence : 0.6704   
## Detection Rate : 0.6598   
## Detection Prevalence : 0.7327   
## Balanced Accuracy : 0.8816   
##   
## '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 Yes   
## 6 No

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 283 92  
## No 33 63  
##   
## Accuracy : 0.7346   
## 95% CI : (0.6923, 0.774)  
## No Information Rate : 0.6709   
## P-Value [Acc > NIR] : 0.00164   
##   
## Kappa : 0.3345   
##   
## Mcnemar's Test P-Value : 2.129e-07   
##   
## Sensitivity : 0.8956   
## Specificity : 0.4065   
## Pos Pred Value : 0.7547   
## Neg Pred Value : 0.6562   
## Prevalence : 0.6709   
## Detection Rate : 0.6008   
## Detection Prevalence : 0.7962   
## Balanced Accuracy : 0.6510   
##   
## 'Positive' Class : Yes   
##