RForestAssignment

Je’Kolby Worthy

2/16/2021

# Module 4

# BAN 502 Random Forest Assignment

### Je’Kolby Worthy

library(tidyverse)

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

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.0.5 ✓ dplyr 1.0.3  
## ✓ 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 0.7.3 ✓ 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(VIM)

## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:recipes':  
##   
## prepare

## The following object is masked from 'package:datasets':  
##   
## sleep

library(skimr)  
library(vip)

##   
## Attaching package: 'vip'

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

library(ranger)  
library(skimr)  
include = FALSE

drug\_data\_1 <- 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\_data\_1) = 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\_data\_1)

## 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\_data\_1[drug\_data\_1 == "CL0"] = "No"  
drug\_data\_1[drug\_data\_1 == "CL1"] = "No"  
drug\_data\_1[drug\_data\_1 == "CL2"] = "Yes"  
drug\_data\_1[drug\_data\_1 == "CL3"] = "Yes"  
drug\_data\_1[drug\_data\_1 == "CL4"] = "Yes"  
drug\_data\_1[drug\_data\_1 == "CL5"] = "Yes"  
drug\_data\_1[drug\_data\_1 == "CL6"] = "Yes"  
drug\_data\_1\_clean = drug\_data\_1 %>% 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\_data\_1\_clean)

## tibble [1,885 × 31] (S3: tbl\_df/tbl/data.frame)  
## $ Age : Factor w/ 6 levels "18\_24","25\_34",..: 3 2 3 1 3 6 4 3 3 5 ...  
## $ Gender : Factor w/ 2 levels "Male","Female": 2 1 1 2 2 2 1 1 2 1 ...  
## $ Education: Factor w/ 9 levels "Under16","At16",..: 6 9 6 8 9 4 8 2 6 8 ...  
## $ Country : Factor w/ 7 levels "USA","NewZealand",..: 7 7 7 7 7 6 1 7 6 7 ...  
## $ Ethnicity: Factor w/ 7 levels "Black","Asian",..: 6 3 3 3 3 3 3 3 3 3 ...  
## $ 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 : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Amphet : Factor w/ 2 levels "Yes","No": 1 1 2 2 2 2 2 2 2 2 ...  
## $ Amyl : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Benzos : Factor w/ 2 levels "Yes","No": 1 2 2 1 2 2 2 2 2 2 ...  
## $ Caff : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Cannabis : Factor w/ 2 levels "No","Yes": 1 2 2 2 2 1 1 1 1 1 ...  
## $ Choc : Factor w/ 2 levels "Yes","No": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Coke : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 1 1 1 1 ...  
## $ Crack : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ecstasy : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Heroin : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ketamine : Factor w/ 2 levels "No","Yes": 1 2 1 2 1 1 1 1 1 1 ...  
## $ Legalh : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LSD : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Meth : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 1 ...  
## $ Mushrooms: Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 1 1 1 ...  
## $ Nicotine : Factor w/ 2 levels "Yes","No": 1 1 2 1 1 1 1 2 1 1 ...  
## $ Semer : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ VSA : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...

drug\_data1\_clean = drug\_data\_1\_clean %>% select(!(Alcohol:Mushrooms)) %>% select(!(Semer:VSA))  
names(drug\_data\_1\_clean)

## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Alcohol" "Amphet" "Amyl" "Benzos" "Caff" "Cannabis"   
## [19] "Choc" "Coke" "Crack" "Ecstasy" "Heroin" "Ketamine"   
## [25] "Legalh" "LSD" "Meth" "Mushrooms" "Nicotine" "Semer"   
## [31] "VSA"

skim(drug\_data1\_clean)

Data summary

|  |  |
| --- | --- |
| Name | drug\_data1\_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**

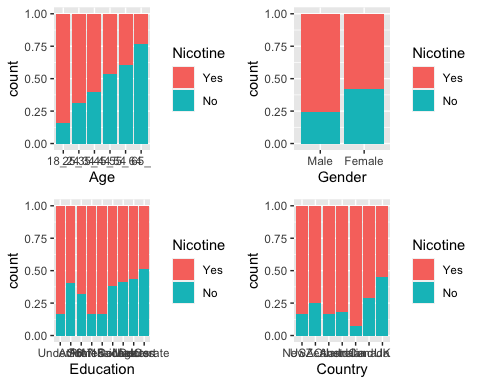
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| Nscore | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.68 | 0.04 | 0.63 | 3.27 | ▁▃▇▅▁ |
| Escore | 0 | 1 | 0.00 | 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 | ▂▇▇▇▅ |

**Task 1:** Check for missing data in our “drug\_clean” dataframe. Is there any missingness? **No, there is no missingness in the dataframe.**

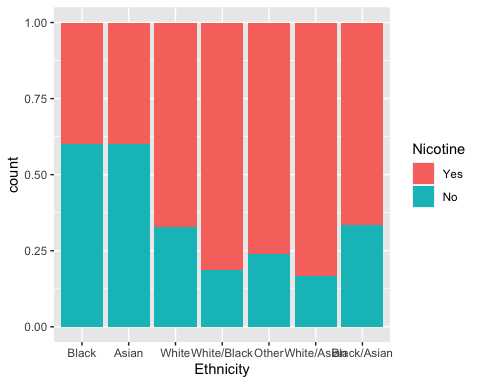
set.seed(1234)  
drug\_data1\_clean\_split = initial\_split(drug\_data1\_clean, prop = 0.7, strata = Nicotine)  
train = training(drug\_data1\_clean\_split)  
test = testing(drug\_data1\_clean\_split)

Nicotine use decreases as age increases. Nicotine is used by more males than females. They are less likely to use nicotine as a person’s education rises. The UK has the fewest users of nicotine. Nicotine is used by blacks and Asians rather than any other ethnicity. Nicotine is more likely to be used by people with high N & O scores. It is less likely that higher Ascores will use it.

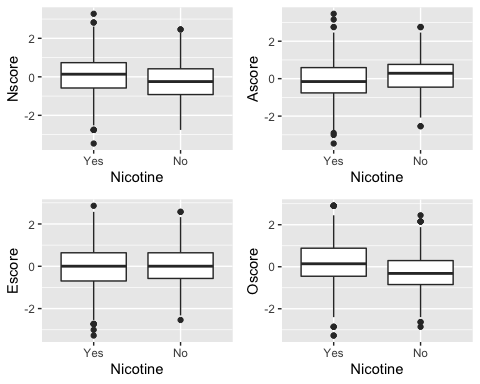
p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill =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



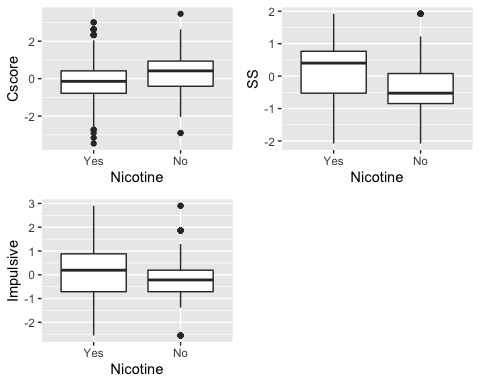
p1 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1)



p1 = ggplot(train, aes(x = Nicotine, y = Nscore)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y = Ascore)) + 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, ncol= 2)



p1 = ggplot(train, aes(x = Nicotine, y = Cscore)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y = SS)) + geom\_boxplot()  
p3 = ggplot(drug\_data\_1\_clean, aes(x = Nicotine, y = Impulsive)) + geom\_boxplot()  
grid.arrange(p1,p2,p3, ncol = 2)



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

drug\_data1\_clean\_split\_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\_data1\_clean\_split\_wflow <-  
 workflow() %>%  
 add\_model(rf\_model) %>%  
 add\_recipe(drug\_data1\_clean\_split\_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\_data1\_clean\_split\_wflow,  
 resamples = rf\_folds,  
 grid = rf\_grid)

##   
## Attaching package: 'rlang'

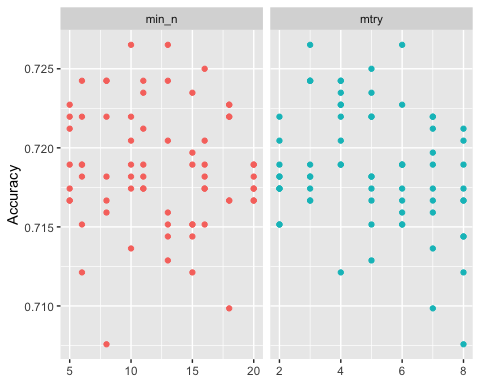
## The following objects are masked from 'package:purrr':  
##   
## %@%, as\_function, flatten, flatten\_chr, flatten\_dbl, flatten\_int,  
## flatten\_lgl, flatten\_raw, invoke, list\_along, modify, prepend,  
## splice

##   
## 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")

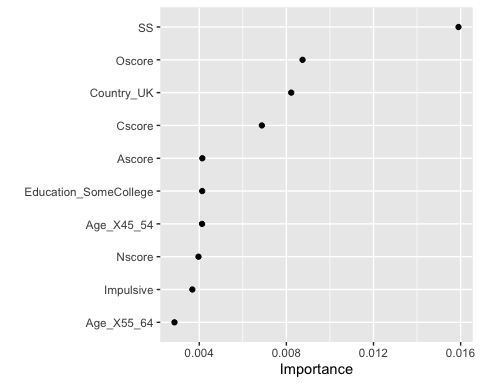


best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 drug\_data1\_clean\_split\_wflow,  
 best\_rf  
)  
final\_rf

## ══ 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 = 10  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

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

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



Task 5: What variables are most important in this model? While looking at this model, the variables that appear to be the most important are SS, followed by Country\_UK, and Oscore.

Task 6: How does the model perform on the training and testing sets? On the training set this model works well, there is a decrease in accuracy when it is performed 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 868 129  
## No 17 306  
##   
## Accuracy : 0.8894   
## 95% CI : (0.8712, 0.9058)  
## No Information Rate : 0.6705   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7322   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9808   
## Specificity : 0.7034   
## Pos Pred Value : 0.8706   
## Neg Pred Value : 0.9474   
## Prevalence : 0.6705   
## Detection Rate : 0.6576   
## Detection Prevalence : 0.7553   
## Balanced Accuracy : 0.8421   
##   
## '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 331 115  
## No 48 71  
##   
## Accuracy : 0.7115   
## 95% CI : (0.6722, 0.7486)  
## No Information Rate : 0.6708   
## P-Value [Acc > NIR] : 0.02111   
##   
## Kappa : 0.2808   
##   
## Mcnemar's Test P-Value : 2.347e-07   
##   
## Sensitivity : 0.8734   
## Specificity : 0.3817   
## Pos Pred Value : 0.7422   
## Neg Pred Value : 0.5966   
## Prevalence : 0.6708   
## Detection Rate : 0.5858   
## Detection Prevalence : 0.7894   
## Balanced Accuracy : 0.6275   
##   
## 'Positive' Class : Yes   
##

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? This model could be used in a real-world example to a particular group that might be likely to smoke. Knowing this group could help to build ads that would inspire these individuals to stop smoking. It’s inconsistency due to over lifting, could be a issue for this model.