1. False
2. 1318
3. True
4. True
5. B. 0.730
6. D. SS
7. 0.9165
8. 0.6707
9. 0.6966
10. over fitting is most likely occurring

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.0.0 ──  
## ✔ broom 1.0.1 ✔ rsample 1.1.0  
## ✔ dials 1.0.0 ✔ tune 1.0.0  
## ✔ infer 1.0.3 ✔ workflows 1.0.0  
## ✔ modeldata 1.0.1 ✔ workflowsets 1.0.0  
## ✔ parsnip 1.0.1 ✔ yardstick 1.1.0  
## ✔ recipes 1.0.1   
## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Dig deeper into tidy modeling with R at https://www.tmwr.org

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)  
library(readr)

drug <- read\_csv("drug\_data-2.csv")

## Rows: 1885 Columns: 32  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (19): Column14, Column15, Column16, Column17, Column18, Column19, Column...  
## dbl (13): Column1, Column2, Column3, Column4, Column5, Column6, Column7, Col...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

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()` 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)

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

Check for Missingness

library(skimr)  
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**

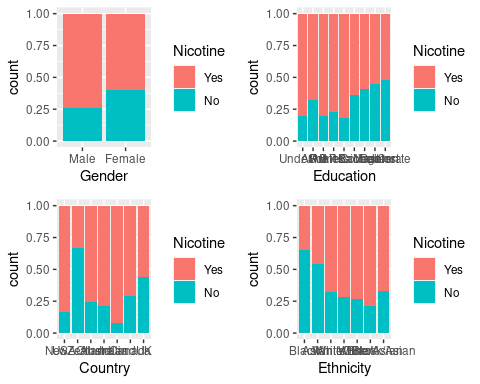
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nscore | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.68 | 0.04 | 0.63 | 3.27 | ▁▃▇▅▁ |
| Escore | 0 | 1 | 0.00 | 1.00 | -3.27 | -0.70 | 0.00 | 0.64 | 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 | ▂▇▇▇▅ |

Split Data into training set (70%) and testing set (30%)

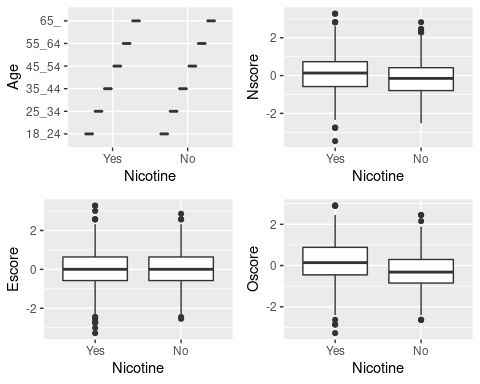
set.seed(1234)  
drug\_split = initial\_split(drug\_clean, prop = 0.70, strata = Nicotine)  
train = training(drug\_split)  
test = testing(drug\_split)

Visualizations (groups of 4)

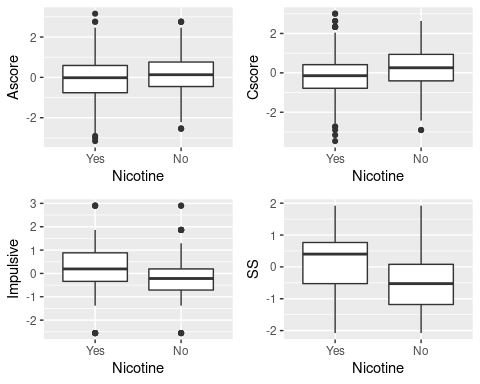
p1 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Ethnicity, fill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



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



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, ncol = 2)



Cross Validation Folds

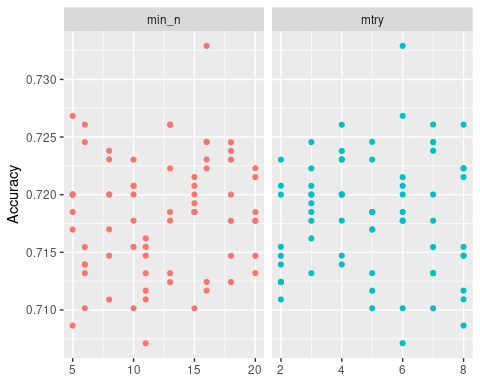
set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

Random Forest Model

drug\_recipe = recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>% #add tuning of mtry and min\_n parameters  
 #setting trees to 100 here should also speed things up a bit, but more trees might be better  
 set\_engine("ranger", importance = "permutation") %>% #added importance metric  
 set\_mode("classification")  
  
drug\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(drug\_recipe)  
  
rf\_grid = grid\_regular(  
 mtry(range = c(2, 8)), #these values determined through significant trial and error  
 min\_n(range = c(5, 20)), #these values determined through significant trial and error  
 levels = 10  
)  
  
set.seed(123)  
rf\_res\_tuned = tune\_grid(  
 drug\_wflow,  
 resamples = rf\_folds,  
 grid = rf\_grid #use the tuning grid  
)

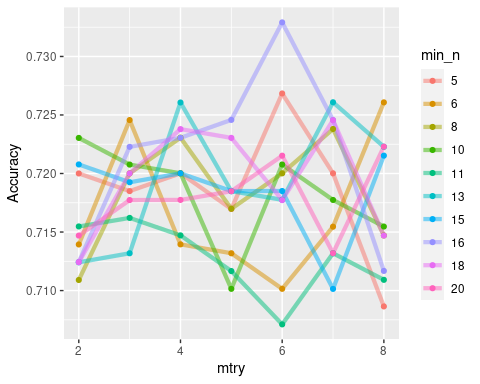
Parameter Performance

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")



Alternate Parameter View

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")



Best mtry and min\_n values fit

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

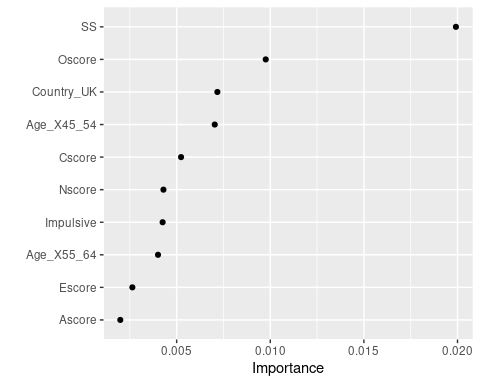
## ══ Workflow ════════════════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 1 Recipe Step  
##   
## • step\_dummy()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 6  
## trees = 100  
## min\_n = 16  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

#fit the finalized workflow to our training data  
final\_rf\_fit = fit(final\_rf, train)

Variable Importance

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

## Warning: `pull\_workflow\_fit()` was deprecated in workflows 0.2.3.  
## Please use `extract\_fit\_parsnip()` instead.



Predictions on Training Set

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

## # A tibble: 6 × 1  
## .pred\_class  
## <fct>   
## 1 No   
## 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 871 97  
## No 13 337  
##   
## Accuracy : 0.9165   
## 95% CI : (0.9003, 0.9309)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8013   
##   
## Mcnemar's Test P-Value : 2.498e-15   
##   
## Sensitivity : 0.9853   
## Specificity : 0.7765   
## Pos Pred Value : 0.8998   
## Neg Pred Value : 0.9629   
## Prevalence : 0.6707   
## Detection Rate : 0.6608   
## Detection Prevalence : 0.7344   
## Balanced Accuracy : 0.8809   
##   
## 'Positive' Class : Yes   
##

Predictions on Testing Set

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

## # A tibble: 6 × 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 327 119  
## No 53 68  
##   
## Accuracy : 0.6966   
## 95% CI : (0.657, 0.7343)  
## No Information Rate : 0.6702   
## P-Value [Acc > NIR] : 0.09699   
##   
## Kappa : 0.2462   
##   
## Mcnemar's Test P-Value : 7.188e-07   
##   
## Sensitivity : 0.8605   
## Specificity : 0.3636   
## Pos Pred Value : 0.7332   
## Neg Pred Value : 0.5620   
## Prevalence : 0.6702   
## Detection Rate : 0.5767   
## Detection Prevalence : 0.7866   
## Balanced Accuracy : 0.6121   
##   
## 'Positive' Class : Yes   
##