Almond\_RandomForest

Ryan Almond

2/11/2022

drug <- read\_csv("drug\_data-1.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")  
  
#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()` 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))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_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"

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 |

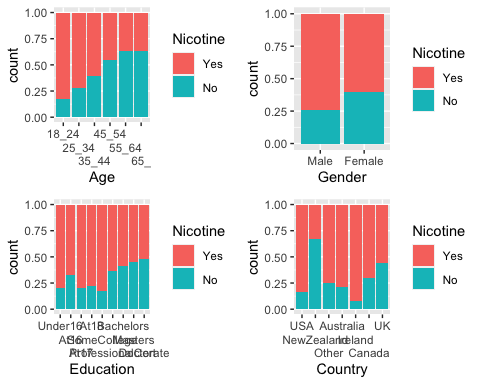
**Variable type: numeric**

| 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 | ▂▇▇▇▅ |

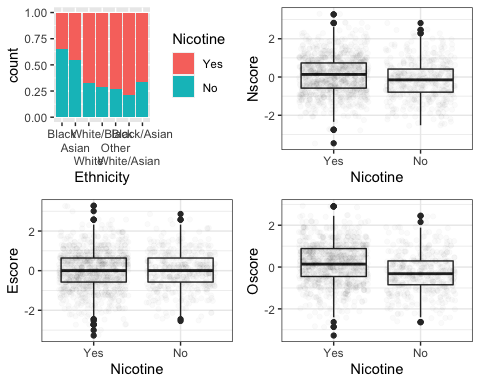
According to the above skim there is no missingness.

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

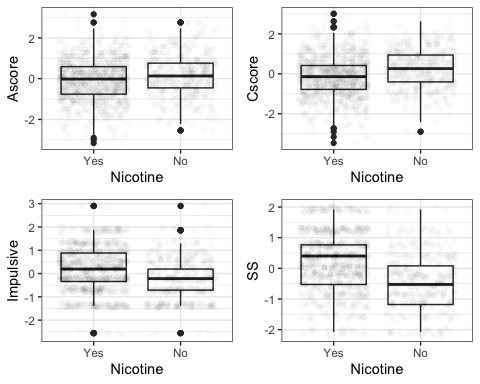
p1 = ggplot(train,aes(x=Age,fill=Nicotine)) + geom\_bar(position = "fill")+scale\_x\_discrete(guide = guide\_axis(n.dodge=3))  
p2 = ggplot(train,aes(x=Gender,fill=Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train,aes(x=Education,fill=Nicotine)) + geom\_bar(position = "fill")+scale\_x\_discrete(guide = guide\_axis(n.dodge=3))  
p4 = ggplot(train,aes(x=Country,fill=Nicotine)) + geom\_bar(position = "fill")+scale\_x\_discrete(guide = guide\_axis(n.dodge=3))  
p5 = ggplot(train,aes(x=Ethnicity,fill=Nicotine)) + geom\_bar(position = "fill")+scale\_x\_discrete(guide = guide\_axis(n.dodge=3))  
p6 = ggplot(train,aes(x=Nicotine,y=Nscore)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
p7 = ggplot(train,aes(x=Nicotine,y=Escore)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
p8 = ggplot(train,aes(x=Nicotine,y=Oscore)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
p9 = ggplot(train,aes(x=Nicotine,y=Ascore)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
p10 = ggplot(train,aes(x=Nicotine,y=Cscore)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
p11 = ggplot(train,aes(x=Nicotine,y=Impulsive)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
p12 = ggplot(train,aes(x=Nicotine,y=SS)) + geom\_boxplot() + geom\_jitter(alpha = 0.02) + theme\_bw()  
grid.arrange(p1,p2,p3,p4, ncol = 2) #arranging ggplot objects in a grid



grid.arrange(p5,p6,p7,p8, ncol = 2) #arranging ggplot objects in a grid



grid.arrange(p9,p10,p11,p12, ncol = 2) #arranging ggplot objects in a grid

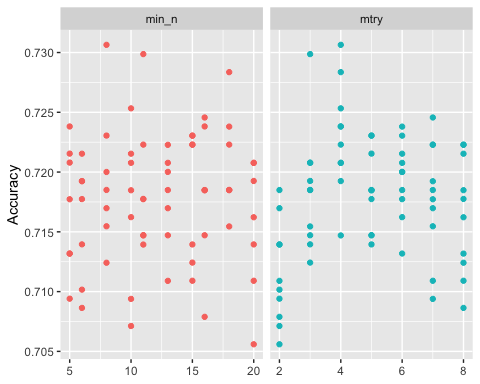


set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

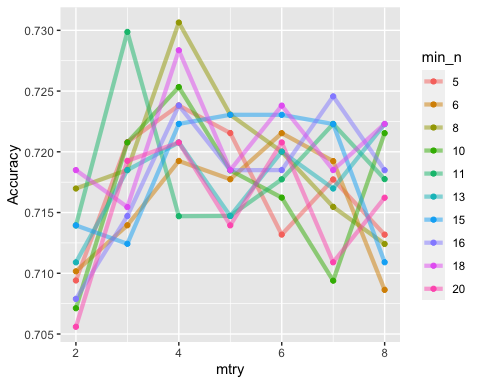
Random forest with an R-defined tuning grid (this model took about 5 minutes to run)

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 #use the tuning 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(alpha = 0.5, size = 1.5) +  
 geom\_point() +  
 labs(y = "Accuracy")



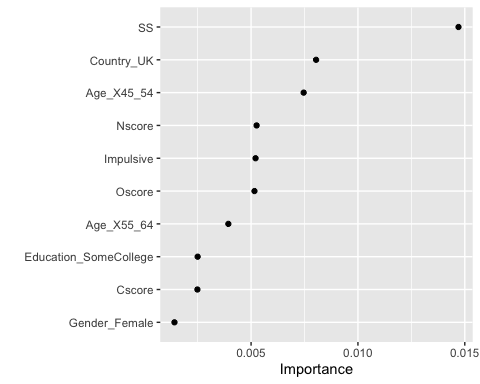
best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 drug\_wflow,  
 best\_rf  
)  
  
final\_rf

## ══ Workflow ════════════════════════════════════════════════════════════════════  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## ── Preprocessor ────────────────────────────────────────────────────────────────  
## 1 Recipe Step  
##   
## • step\_dummy()  
##   
## ── Model ───────────────────────────────────────────────────────────────────────  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 4  
## trees = 100  
## min\_n = 8  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

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

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.



We can see that SS(sensation seeing measured by ImpSS),Country\_UK,Age\_X45\_54,Nscore(NEO-FFI-R Neuroticism),Impulsive(impulsiveness measured by BIS-11),Oscore(NEO-FFI-R Openness to experience) are the most important.

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 879 78  
## No 5 356  
##   
## Accuracy : 0.937   
## 95% CI : (0.9225, 0.9495)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8511   
##   
## Mcnemar's Test P-Value : 2.722e-15   
##   
## Sensitivity : 0.9943   
## Specificity : 0.8203   
## Pos Pred Value : 0.9185   
## Neg Pred Value : 0.9861   
## Prevalence : 0.6707   
## Detection Rate : 0.6669   
## Detection Prevalence : 0.7261   
## Balanced Accuracy : 0.9073   
##   
## 'Positive' Class : Yes   
##

It works great on the training set with an accuracy of 0.9416.

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

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 332 115  
## No 48 72  
##   
## Accuracy : 0.7125   
## 95% CI : (0.6733, 0.7495)  
## No Information Rate : 0.6702   
## P-Value [Acc > NIR] : 0.01708   
##   
## Kappa : 0.2846   
##   
## Mcnemar's Test P-Value : 2.347e-07   
##   
## Sensitivity : 0.8737   
## Specificity : 0.3850   
## Pos Pred Value : 0.7427   
## Neg Pred Value : 0.6000   
## Prevalence : 0.6702   
## Detection Rate : 0.5855   
## Detection Prevalence : 0.7884   
## Balanced Accuracy : 0.6294   
##   
## 'Positive' Class : Yes   
##

It is not quite as good on the test set but still decent with an accuracy of 0.709.

I think that it can be used to see who to target for potential rehabilitation and education. We can see with this specifically with nicotine and a lot of younger people and specific race groups that could be targeted. People who fall on specific sections of the questionnaires and exams can also be targeted. This was only nicotine which is definitely on the lower end of harm compared to the other drugs in the original data set. The other drugs can follow the same thing we did here to see who is the most at risk and try to take preventative actions. I personally do not like putting people into specific categories and targeting them because of the category since there are definite outliers. If we are talking about drugs as a whole there is a stigma around them that makes it harder for people to seek help and I think models such as this would only further it. To make an actual change on drug usage there needs to be systemic change. We are already 0-1 on the war on drugs.