# Module 4 - Assignment 2 - Random Forests

## Samantha Carter

Task 1: Is there any missingness? No

Task 3: Comment on the relationship between each variable and “Nicotine”.  
\* Age seems to affect nicotine as age goes up so does nicotine use.   
\* Females seems to use nicotine more than men.   
\* Education affects it a little, rising with more education (This might also correlate with age since people get more educated as they get older.)   
\* Country it’s pretty clear being from the UK affects nicotine use. It’s much higher than other countries even accounting for 55% of the survey’s participant coming from the UK.   
\* Ethnicity Black seems to have a higher likelihood.   
\* Extrovert score doesn’t affect nicotine.   
\* Neuroticism score doesn’t affect nicotine.   
\* Openness score seems to have an affect with yes being more likely to use nicotine.   
\* Agreeableness score doesn’t affect nicotine.   
\* Impulsiveness those individuals that are impulsive seem more likely to use nicotine   
\* Sensation seeking those that are sensation seeking are far more likely to use nicotine this variable definitely affects nicotine use   
\* Conscientiousness score affects nicotine use a little. Not conscientious seems to make it more likely that a participant will use nicotine.

Task 5: What variables are most important in this model? The most important variable to predict whether someone uses Nicotine is sensation seeking, which logically makes sense. Follwed by whether or not you live in the UK remembering 55% of the data is participants living in the UK. And the 3rd is their openness to experience score, which also makes a lot of sense those not open to new experiences would probably never try nicotine in the first place.

Task 6: How does the model perform on the training and testing sets? Training 0.93 Testing 0.71

Task 7:Comment on how this model might be used in the “real-world.” Would you recommend this model for real-world use? No What if any concerns would you have about using the model?, It seems the model we developed from the training set is overfit since the training set is 93% accuracy and the testing set has a much lower 71% accuracy.I would definitely try to find a better predictive model.

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()` 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))

#summary(drug\_clean)  
#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

str(drug\_clean)

## tibble[,13] [1,885 x 13] (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 ...  
## $ Nicotine : Factor w/ 2 levels "Yes","No": 1 1 2 1 1 1 1 2 1 1 ...

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   
##

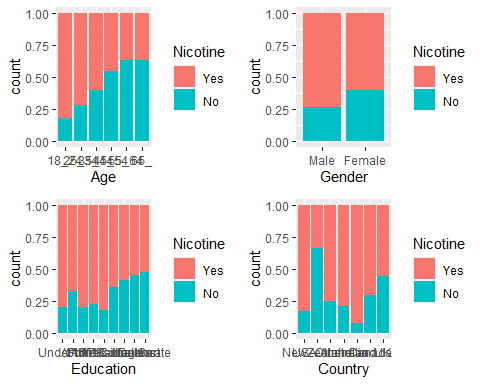
#view(drug\_clean)

#### Task 2

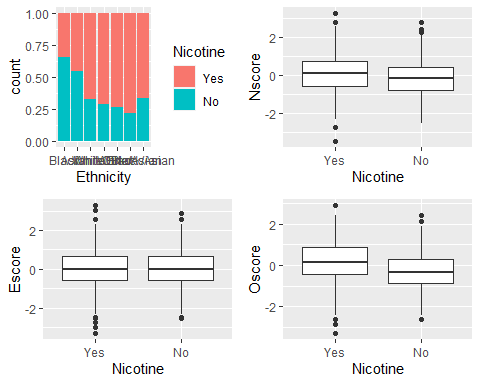
set.seed(1234, kind = "Mersenne-Twister", normal.kind = "Inversion")  
drug\_clean\_split = initial\_split(drug\_clean, prop = 0.70, strata = "Nicotine" )  
train = training(drug\_clean\_split)  
test = testing(drug\_clean\_split)

#### Task 3

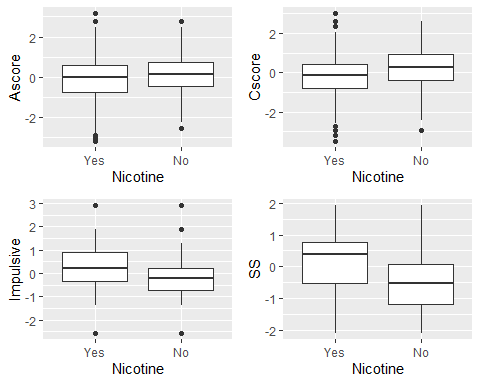
p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill =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

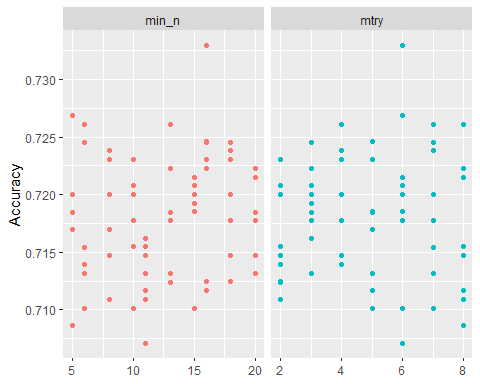


#### Task 4

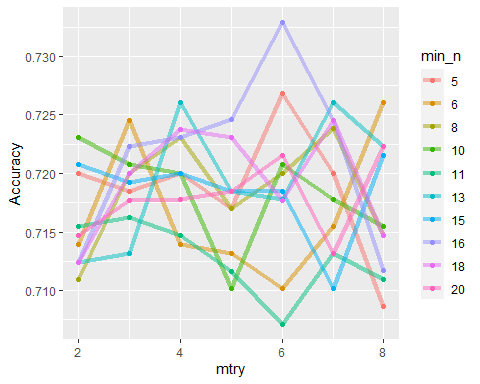
set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

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



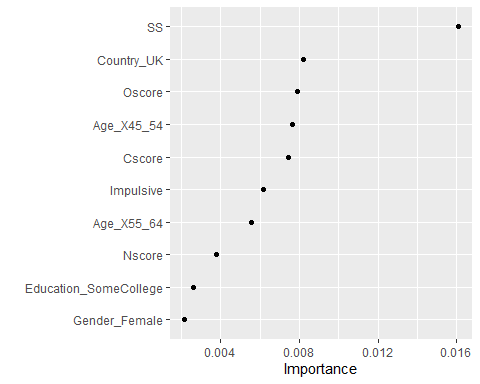
#### Task 5

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

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

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



#### Task 6

trainpredrf = predict(final\_rf\_fit, train)  
#head(trainpredrf)  
confusionMatrix(trainpredrf$.pred\_class, train$Nicotine,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 870 92  
## No 14 342  
##   
## Accuracy : 0.9196   
## 95% CI : (0.9036, 0.9337)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8092   
##   
## Mcnemar's Test P-Value : 7.495e-14   
##   
## Sensitivity : 0.9842   
## Specificity : 0.7880   
## Pos Pred Value : 0.9044   
## Neg Pred Value : 0.9607   
## Prevalence : 0.6707   
## Detection Rate : 0.6601   
## Detection Prevalence : 0.7299   
## Balanced Accuracy : 0.8861   
##   
## 'Positive' Class : Yes   
##

Predictions on test

testpredrf = predict(final\_rf\_fit, test)  
#head(testpredrf)  
confusionMatrix(testpredrf$.pred\_class, test$Nicotine,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 329 118  
## No 51 69  
##   
## Accuracy : 0.7019   
## 95% CI : (0.6624, 0.7393)  
## No Information Rate : 0.6702   
## P-Value [Acc > NIR] : 0.05808   
##   
## Kappa : 0.2583   
##   
## Mcnemar's Test P-Value : 3.836e-07   
##   
## Sensitivity : 0.8658   
## Specificity : 0.3690   
## Pos Pred Value : 0.7360   
## Neg Pred Value : 0.5750   
## Prevalence : 0.6702   
## Detection Rate : 0.5802   
## Detection Prevalence : 0.7884   
## Balanced Accuracy : 0.6174   
##   
## 'Positive' Class : Yes   
##