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)

## spec\_tbl\_df [1,885 x 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()  
## .. )  
## - attr(\*, "problems")=<externalptr>

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)

## tibble [1,885 x 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))  
names(drug\_clean)

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

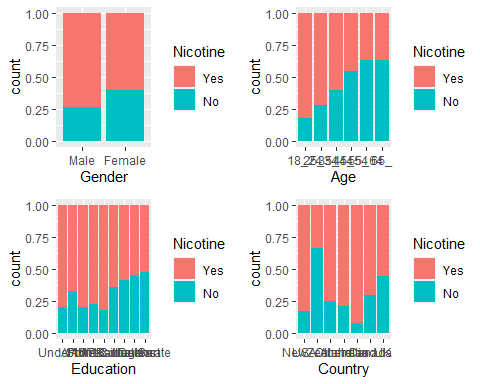
#summary(drug\_clean)

There does not appear to be any missingness remaining, but I did contemplate “Other” in education. However, that is still a valid answer to the category, so I will leave it be. The remaining rows all have a selection or fit within the confines of the quantitative variables.

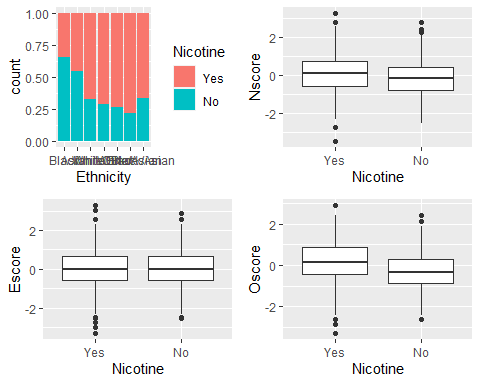
set.seed(1234)   
drug\_split = initial\_split(drug\_clean, prop = 0.7, strata = Nicotine) #70% in training  
train = training(drug\_split)  
test = testing(drug\_split)

Let’s do some visualizations!

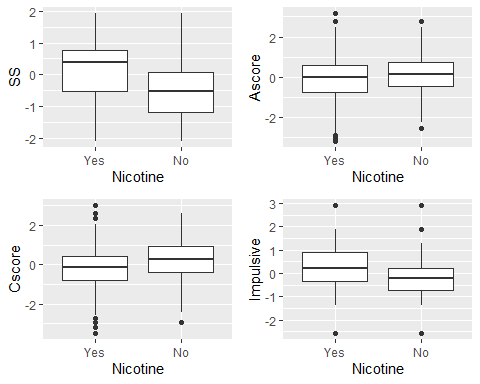
p1 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill =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

 Being male appears to increase your chance of using nicotine, as does being younger. Although that could be attributed to new trends in smoking marketing or initial addictions from young people who don’t know any better/contemplate the ramifications of their actions! Having less education also increases your chances of using Nicotine, as does being in any country other than NZ and possibly the UK.

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, ncol =2)

 Looking at ethnicity, being “White”, “White/Black”, “Other” or “White/Asian” appears to increase your chance of nicotine usage. Moreover, being neurotic increases the odds as does Open to new experiences. Extraversion does not appear to have a major effect.

p1 = ggplot(train, aes(x = Nicotine, y = SS)) + geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y= Ascore)) + geom\_boxplot()  
p3 = ggplot(train, aes(x = Nicotine, y = Cscore)) + geom\_boxplot()  
p4 = ggplot(train, aes(x= Nicotine, y = Impulsive)) + geom\_boxplot()  
grid.arrange(p1,p2,p3,p4, ncol =2)

 Lastly, being sensitive (SS)/ insensed obviously provides the punch for being inclined towards nicotine usage. AScore i.e. agreeableness seems to not tell a compelling correaltion/relationship, as does Cscore i.e. conscientiousness, However, impulsiveness does lean itself towards nicotine usage, which is to be expected as it is very much an impulse habit.

Now, let’s make a random forest model on training to predict Nicotine!

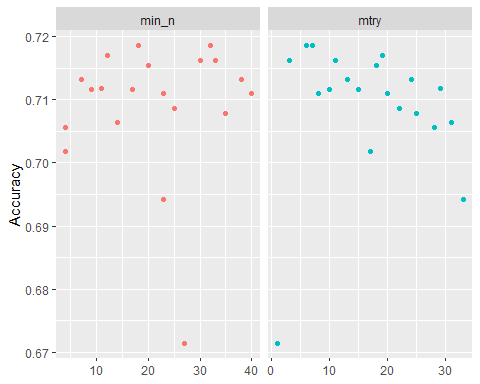
Set up our folds for cross-validation

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

drug\_recipe = recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
dr\_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(dr\_model) %>%   
 add\_recipe(drug\_recipe)  
  
dr\_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) #make sure seed is here  
rf\_res = tune\_grid(  
 drug\_wflow,  
 resamples = rf\_folds,  
 grid = 20 #try 20 different combinations of the random forest tuning parameters  
)

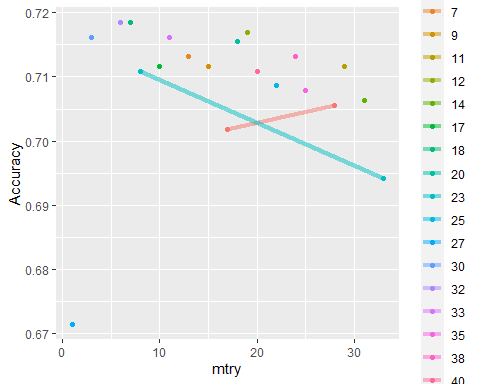
## i Creating pre-processing data to finalize unknown parameter: mtry

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")



As we increase N try, we see a slight increase in accuracy. As we increase mtry, we see a slight decrease in accuracy. If we use larger N and lower mtry, that might be the right way to go!

rf\_res %>%  
 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")



Let’s try to find the most accurate measures here.

best\_rf = select\_best(rf\_res, "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 = 7  
## trees = 100  
## min\_n = 18  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

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

It seems mtry of 7 and min\_n of 18 seem to provide the most accurate results.

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

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 Yes   
## 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 868 111  
## No 16 323  
##   
## Accuracy : 0.9036   
## 95% CI : (0.8864, 0.919)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.769   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9819   
## Specificity : 0.7442   
## Pos Pred Value : 0.8866   
## Neg Pred Value : 0.9528   
## Prevalence : 0.6707   
## Detection Rate : 0.6586   
## Detection Prevalence : 0.7428   
## Balanced Accuracy : 0.8631   
##   
## 'Positive' Class : Yes   
##

Looking at the accuracy of aboe 90% (90.36%) alongside a lesser naive accuracy + p-value, all of this points to at least being more comfortable and having some reliability in this model.

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

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

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

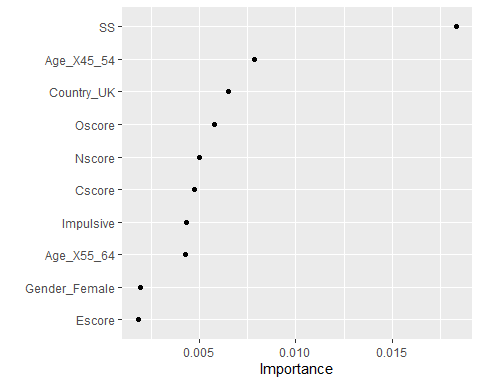
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 324 117  
## No 56 70  
##   
## Accuracy : 0.6949   
## 95% CI : (0.6552, 0.7326)  
## No Information Rate : 0.6702   
## P-Value [Acc > NIR] : 0.1135   
##   
## Kappa : 0.2475   
##   
## Mcnemar's Test P-Value : 5.074e-06   
##   
## Sensitivity : 0.8526   
## Specificity : 0.3743   
## Pos Pred Value : 0.7347   
## Neg Pred Value : 0.5556   
## Prevalence : 0.6702   
## Detection Rate : 0.5714   
## Detection Prevalence : 0.7778   
## Balanced Accuracy : 0.6135   
##   
## 'Positive' Class : Yes   
##

The test accuracy is not as great (just below 70%) and the naive rate is only slightly below that at 67%. The p-value is >.05 which does not instill a lot of confidence either. The model doesn’t appear to perform as well on the testing set as it does with the training set.

Because of this, I would hesitate a bit before applying this model to real-world scenarios where the model could be used to analyze or make critical decisions regarding public health, for example. While the training set does have high accuracy, the discrepancy with the testing model would make me careful before recommending it to the general data-driven public. Granted, I do think there is still value here in terms of utilizing this to help find traits, habits, connections with subjects and nicotine use, but it does draw someone concerns that there is not as much parity between the training/testing sets.

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.



It appears SS, Being 45-54, from the UK, and having a higher “openness to experience” score all collate together to being inclined towards nicotine usage. The age demographic makes sense, as smoking was a generational habit, as does being more open to experiences would naturally push one towards that as well. SS referring to those who are sensation seeking, which would make perfect sense in identifying factors that would obviously indicate who would be a likely or currently nicotine user/seeker.