;

# Strickland,, Marie

## BAN 502 - Predictive Analytics

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))

#str(drug\_clean)

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

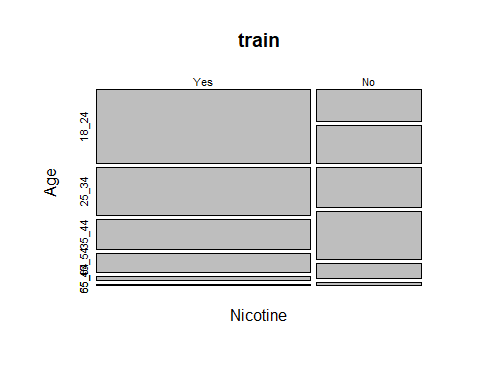
## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Nicotine"

#summary(drug\_clean)

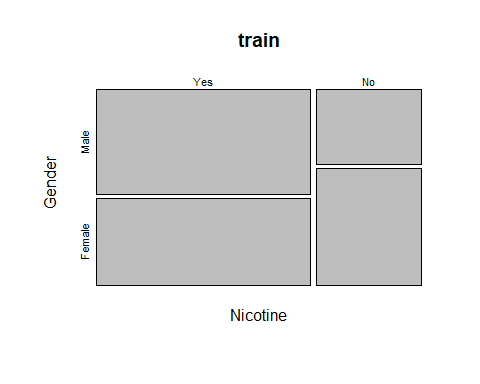
There is no missingness in the data set.

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

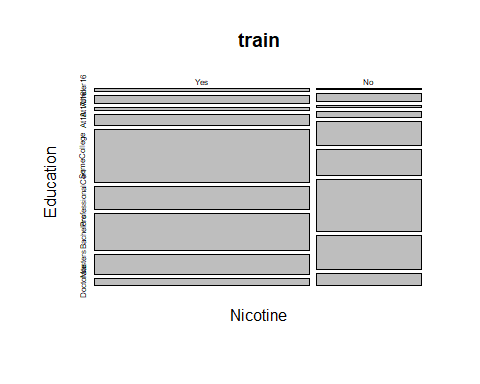
mosaicplot(Nicotine~Age, train)



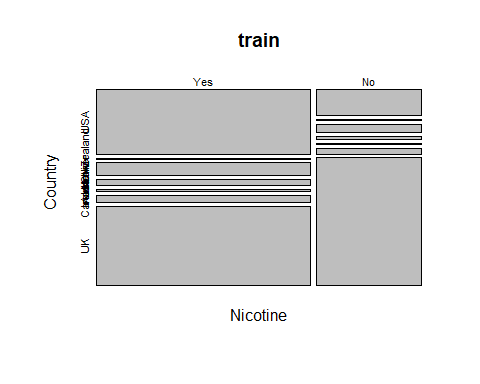
mosaicplot(Nicotine~Gender, train)



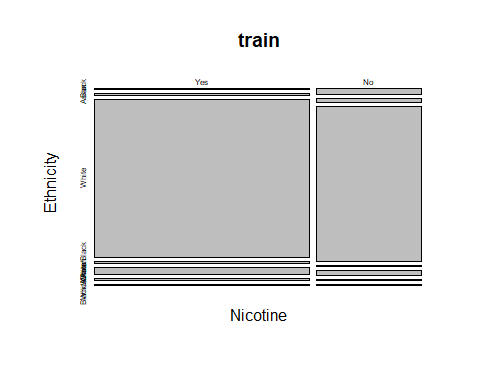
mosaicplot(Nicotine~Education, train,  
 cex.axis = 0.5)



mosaicplot(Nicotine~Country, train)

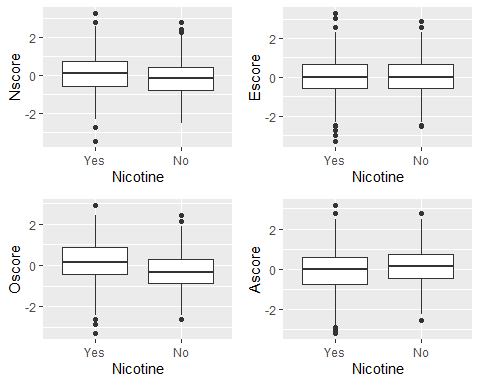


mosaicplot(Nicotine~Ethnicity, train,  
 cex.axis = 0.5)

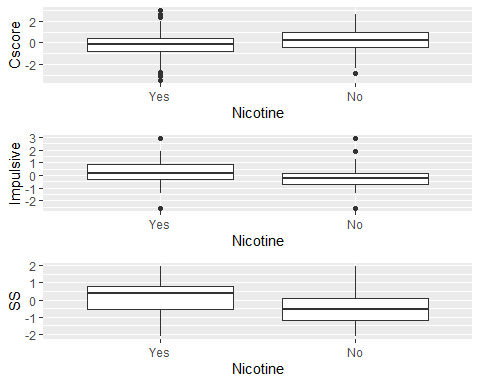


I did mosaic plots for the factor variables. UK and US are more likely to smoke nicotine. Education seems like s good predictor variable especially if only did some college. Females are more likely to not smoke than males. The younger you are the more likely you are to smoke. Ethnicity does not seem like a great predictor as there is little difference between the number of those who smoke and don’t across each ethnicity.

p1 = ggplot(train, aes(x = Nicotine, y = Nscore)) +  
 geom\_boxplot()  
p2 = ggplot(train, aes(x = Nicotine, y = Escore)) +  
 geom\_boxplot()  
p3 = ggplot(train, aes(x = Nicotine, y = Oscore)) +  
 geom\_boxplot()  
p4 = ggplot(train, aes(x = Nicotine, y = Ascore)) +  
 geom\_boxplot()  
p5 = ggplot(train, aes(x = Nicotine, y = Cscore)) +  
 geom\_boxplot()  
p6 = ggplot(train, aes(x = Nicotine, y = Impulsive)) +  
 geom\_boxplot()  
p7 = ggplot(train, aes(x = Nicotine, y = SS)) +  
 geom\_boxplot()  
grid.arrange(p1, p2, p3, p4)



grid.arrange(p5, p6, p7)

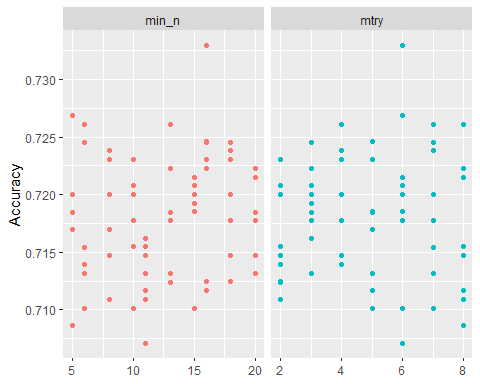


Nscore is for neuroticism, and smokers have a bigger range of scores, but it doesn’t seem that smokers are more likely to have a higher score. Extraversion (Escore) is similar, smokers and non-smokers have the same average of 0. Oscore seems like a little better predictor, which would make sense to me that if you smoke you are more open to do other things. sensation seeking seems like the best predictor as smokers have a higher mean score.

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\_wf = 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 = tune\_grid(  
 drug\_wf,  
 resamples = rf\_folds,  
 grid = rf\_grid  
)

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")



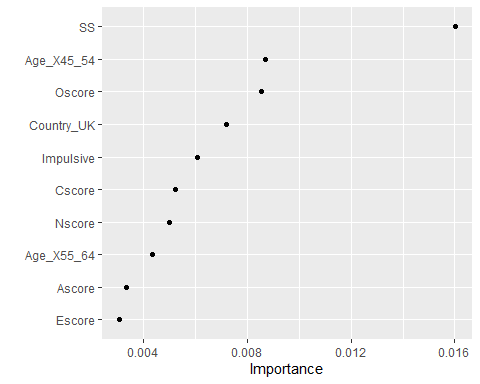
drug\_recipe = recipe(Nicotine~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>%  
 set\_engine("ranger", importance = "permutation") %>%  
 set\_mode("classification")  
  
drug\_wf = workflow() %>%  
 add\_model(rf\_model) %>%  
 add\_recipe(drug\_recipe)  
  
rf\_grid = grid\_regular(  
 mtry(range = c(6,6)),  
 min\_n(range = c(16,17)),  
 levels = 10  
)  
  
set.seed(123)  
rf\_res = tune\_grid(  
 drug\_wf,  
 resamples = rf\_folds,  
 grid = rf\_grid  
)

best\_rf = select\_best(rf\_res, "accuracy")  
  
final\_rf = finalize\_workflow(  
 drug\_wf,  
 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 = 17  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

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

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



SS is most important, which is sensation seeking. UK is 2nd most important which is interesting.

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 866 103  
## No 18 331  
##   
## Accuracy : 0.9082   
## 95% CI : (0.8913, 0.9232)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7813   
##   
## Mcnemar's Test P-Value : 2.234e-14   
##   
## Sensitivity : 0.9796   
## Specificity : 0.7627   
## Pos Pred Value : 0.8937   
## Neg Pred Value : 0.9484   
## Prevalence : 0.6707   
## Detection Rate : 0.6571   
## Detection Prevalence : 0.7352   
## Balanced Accuracy : 0.8712   
##   
## 'Positive' Class : Yes   
##

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

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 866 103  
## No 18 331  
##   
## Accuracy : 0.9082   
## 95% CI : (0.8913, 0.9232)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7813   
##   
## Mcnemar's Test P-Value : 2.234e-14   
##   
## Sensitivity : 0.9796   
## Specificity : 0.7627   
## Pos Pred Value : 0.8937   
## Neg Pred Value : 0.9484   
## Prevalence : 0.6707   
## Detection Rate : 0.6571   
## Detection Prevalence : 0.7352   
## Balanced Accuracy : 0.8712   
##   
## 'Positive' Class : Yes   
##

The model did really well on both the training and testing sets, with 92.26% accuracy for both.

I think this could work in the real-world. It can be used to predict future prices or costs of something or predict a company’s revenue or profit.

I think relying too much on predictions can be dangerous so if using this know that there is always the chance of randomness.