M4 Assignment 2

MTracka

6/6/2021

**#Dataset preparation**

#a) libraries upload

#b) loading the data and adding columns names using function ‘names’

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)

## 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()  
## .. )

#After loading the data and prior the name changing command execution ‘view(drug)’ was used to verify the data. The ‘str(drug)’ command was used to exaamine the struccture of the data set “drug”. The commands were left out from kniting to safe space per assignment instructions.

#c) Change of values: CL0 and CL1 to “No” and CL2, CL3, CL4,CL5, and CL6 to “Yes”

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"

#d) Factor conversion and recoding. To target specific ranges of variables ‘mutate\_at’

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)

# The command ‘str(drug\_clean)’ was used to check the cleaned data and removed before knitting to safe space as per assignment instructions.

#e) Preparation of the set for analysis of the variable “Nicotine” by removal of the remining drug use variables

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"

# There are 13 variables left: “Age”, “Gender”, “Education”, “Country”, “Ethnicity”, “Nscore”, “Escore”, “Oscore”, “Ascore”, “Cscore”, “Impulsive”, “SS”, and “Nicotine”

# Assignment 2 Execution

# #Task 1

#Missing data check and reasonable strategy implementation to address it.

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 | ▂▇▇▇▅ |

# There is no missing data in the “drug\_clean” dataframe.

**#Task 2**

# Data set split into training (70%) and testing (30%) sets, set.seed of 1234, strata - variable “Nicotine”

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

**#Task 3**

# Visualization to examine the relationships between each variable and “Nicotine” variable.

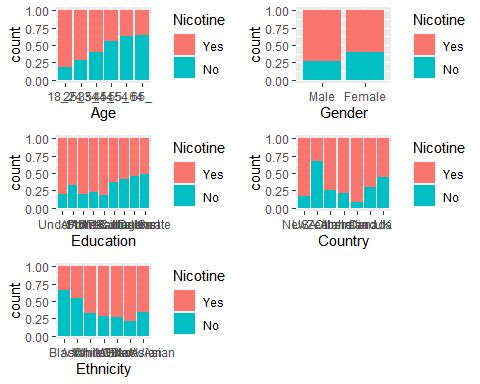
#Summary of the dataframe to allow better readability of variables on the graphs

#Legend: #Nscore - Neuroticism score, Escore-Extraversion score, Oscore- Openness to experience score, Ascore- Agreeableness score, Cscore- Conscientiousness score, Impulsive- Impulsiveness score, SS- sensation seeing score

# summary(drug\_clean)

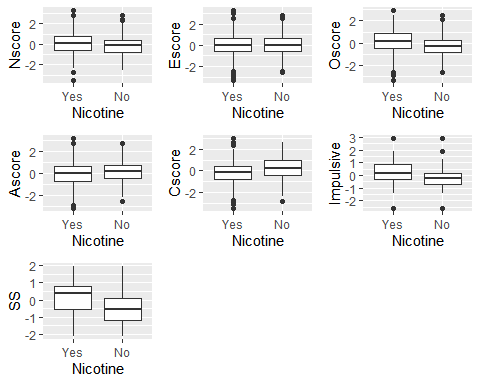
# # Visualization of factor variables

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")  
p5 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill")  
  
grid.arrange(p1,p2,p3,p4,p5)



# Visualization of numeric variables

p5 = ggplot(train, aes(x = Nicotine, y= Nscore)) + geom\_boxplot()  
p6 = ggplot(train, aes(x = Nicotine, y= Escore)) + geom\_boxplot()  
p7 = ggplot(train, aes(x = Nicotine, y= Oscore)) + geom\_boxplot()  
p8 = ggplot(train, aes(x = Nicotine, y= Ascore)) + geom\_boxplot()  
p9 = ggplot(train, aes(x = Nicotine, y= Cscore)) + geom\_boxplot()  
p10 = ggplot(train, aes(x = Nicotine, y= Impulsive)) + geom\_boxplot()  
p11 = ggplot(train, aes(x = Nicotine, y= SS)) + geom\_boxplot()  
grid.arrange(p5,p6,p7,p8,p9,p10,p11)



#Variables such as gender, age group, education, ethnicity, and country impact the Nicotine variable. Based on the data provided females are less likely to use nicotine in comparison to male. The use of nicotine also gets lower with age. The trend of lower use of nicotine correlates to increasing level of eduction. Etnicity is also contributing factor to the use of nicotine as a drug. The use of nicotine in ethnic groups Black, Asian, and Black/Asian is lower, less people from those groups are using nicotine. The use of nicotine by country varies.

# Among the numeric variables SS (sensation seeing score) correllates to the higher nicotine use. Also, Oscore (Openness to experience score) and Impulsive varables are correlated to nicotine use.

**#Task 4**

# Random forest model on training set to predict Nicotine, all variables. Cross-validation 5-fold (set.seed 123.), mtry values 2-8 and min\_n values 5 to 20. Function ‘grid\_regular’ 10 levels . Function ‘tune\_grid’ with random number seed 123. 100 trees.

# Random Forest

drug\_clean\_recipe = recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest() %>%   
 set\_engine("ranger", importance = "permutation") %>%   
 set\_mode("classification")  
  
drug\_clean\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(drug\_clean\_recipe)  
  
set.seed(123)  
drug\_clean\_fit = fit(drug\_clean\_wflow, train)

# Predictions on train set

trainpredrf = predict(drug\_clean\_fit, train)  
head(trainpredrf)

## # A tibble: 6 x 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 880 66  
## No 4 368  
##   
## Accuracy : 0.9469   
## 95% CI : (0.9334, 0.9584)  
## No Information Rate : 0.6707   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8752   
##   
## Mcnemar's Test P-Value : 3.079e-13   
##   
## Sensitivity : 0.9955   
## Specificity : 0.8479   
## Pos Pred Value : 0.9302   
## Neg Pred Value : 0.9892   
## Prevalence : 0.6707   
## Detection Rate : 0.6677   
## Detection Prevalence : 0.7178   
## Balanced Accuracy : 0.9217   
##   
## 'Positive' Class : Yes   
##

# Predictions on test

testpredrf = predict(drug\_clean\_fit, test)  
head(testpredrf)

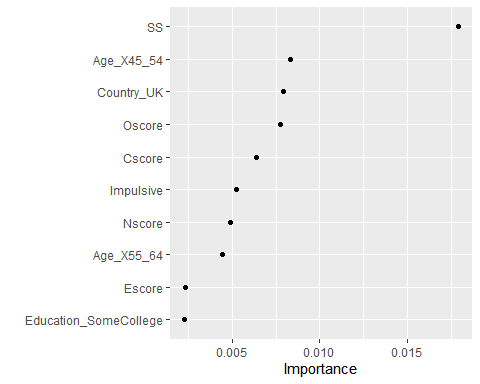
## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 No   
## 2 Yes   
## 3 No   
## 4 No   
## 5 Yes   
## 6 Yes

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 328 118  
## No 52 69  
##   
## Accuracy : 0.7002   
## 95% CI : (0.6606, 0.7376)  
## No Information Rate : 0.6702   
## P-Value [Acc > NIR] : 0.0694   
##   
## Kappa : 0.255   
##   
## Mcnemar's Test P-Value : 6.187e-07   
##   
## Sensitivity : 0.8632   
## Specificity : 0.3690   
## Pos Pred Value : 0.7354   
## Neg Pred Value : 0.5702   
## Prevalence : 0.6702   
## Detection Rate : 0.5785   
## Detection Prevalence : 0.7866   
## Balanced Accuracy : 0.6161   
##   
## 'Positive' Class : Yes   
##

# # Check out variable importance

drug\_clean\_fit %>% pull\_workflow\_fit() %>% vip(geom = "point")



# Set up our folds for cross-validation

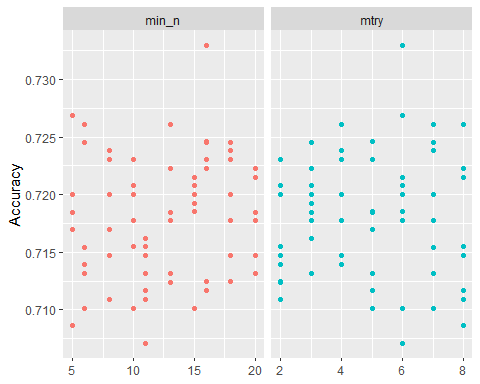
set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

# Tuning grid with assignment specified values for mtry (2-8) and min\_n (5-20)

drug\_clean\_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\_clean\_wflow =   
workflow() %>%   
add\_model(rf\_model) %>%   
add\_recipe(drug\_clean\_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\_clean\_wflow,  
resamples = rf\_folds,  
grid = rf\_grid   
)

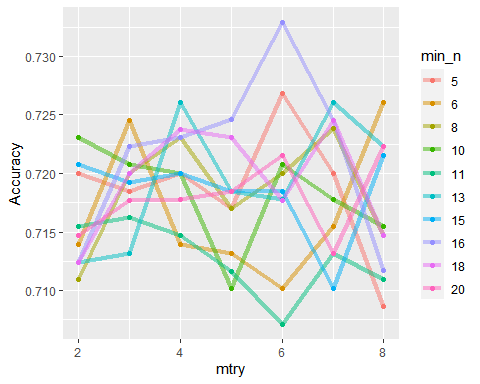
#Visualization of the relationships between parameters and performance metrics #Parameter performance (borrowed from <https://juliasilge.com/blog/sf-trees-random-tuning/>)

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")



#An alternative view of the parameters

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**

#Final model based on the mtry and min\_n from the Task 4. Variable importance.

# Finalization of the workflow

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

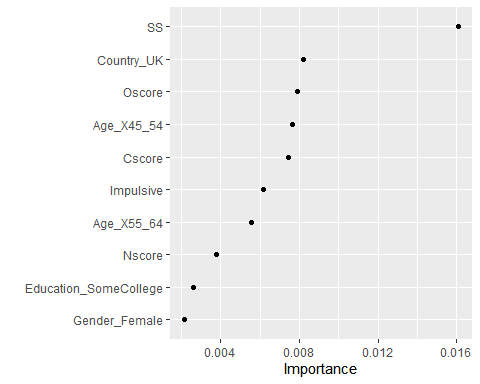
## == Workflow ====================================================================  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## -- Preprocessor ----------------------------------------------------------------  
## 1 Recipe Step  
##   
## \* step\_dummy()  
##   
## -- Model --------------------------------------------------------------------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

# # The workflow fit to train data

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

# Check out variable importance

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



# The most important variable is SS (sensation seeing score), followed by Oscore (Openness to experience score) and Age\_ range \_X45\_54, then Country\_UK, Age \_X55\_64 and Impulsive varables.

# #Task 6

# #The model performance on the train and test sets

# # Predictions on test set

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

# Confusion matrix

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 set

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 No

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   
##

# The performance of the tuned model on the train set is only slightly lower than the model prior tuning (Accuracy of 0.9105 vs 0.9469). Both models perform similarly on the test set (0.7001 vs. 0.6966). In both cases the models are performining better than naive model (0.6707- train set and 0.6702- test set).

# #Task 7

# The tuned model is performing well so it can be used in real world. The tuning improved the variables importance distribution to be more evenly spread. The tuned model is more reliable as it takes into account more than just the main dominant variable.