Random Forests

Libraries: For this assignment you will need the following libraries: tidyverse, tidymodels, caret, gridExtra, vip, and ranger.

Read in the “drug_data.csv” dataset. This dataset deals with drug consumption in individuals across a wide spectrum of countries, drugs, ages, etc. A description of the dataset is available here: http://archive.ics.uci.edu/ml/datasets/Drug+consumption+%28quantified%29

This dataset (as is common with medical and healthcare datasets) requires quite a bit of cleaning and preparation before analysis.

I’ll walk you through the cleaning before we start our random forest work.

Loading the Data

The columns do not have names, so we’ll supply names via the names function.

```
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")
```

Next-up we change all CL0 and CL1 values to “No” and CL2, CL3, CL4, CL5, and CL6 values to “Yes”. CL0 and CL1 imply the drug was never used or used over a decade ago. CL2 through CL6 imply more recent drug use. The code below finds any CL0, CL1, etc. values in the data frame and replaces them with the appropriate “No” or “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"
```

Next up we do a good bit of factor conversion and recoding. Note the use of mutate_at to target specific ranges of variables. You may see a warning message about the use of the funs() function. It is OK to ignore this.

```
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)
```

Take a peek at the cleaned data to make sure we are all good before proceeding.

```
str(drug_clean)
```

We'll focus on Nicotine use, so let's get rid of the remaining drug use variables. We'll use select for this.

```
drug_clean = drug_clean %>% select(!(Alcohol:Mushrooms)) %>% select(!(Semer:VSA))
```

Now we're in business. Let's get started with the tasks.

Question 1: Check for missing data in our "drug_clean" dataframe.

True/False: There is missingness in the dataset.

Question 2: Split the dataset into training (70%) and testing (30%) sets. Use a set.seed of 1234. Stratify by the "Nicotine" variable.
How many rows are in the training set?

Question 3: Create appropriate visualizations (12 in all) to examine the relationships between each variable and "Nicotine". Use grid.arrange (from the gridExtra package) to organize these visuals (perhaps in groups of four visualizations?).

True/False: Individuals in the 18-24 age group are proportionally more likely to be Nicotine users than not.

Question 4: True/False: Individuals with higher "Impulsive" scores more likely to be Nicotine users than not.

Question 5: Create a random forest model (using the ranger package) on the training set to predict Nicotine using all of the variables in the dataset. You 5-fold, k-fold cross-validation (random number seed of 123 for the folds). Allow R to select mtry values between 2 and 8 and min_n values between 5 and 20. Use 10 levels in your "grid_regular" function. Set a random number seed of 123 for the tune_grid function. Use 100 trees.

NOTE: This model may take a few minutes to run. Be patient :)

Visualize the relationships between parameters and performance metrics.

The highest accuracy in this visualization is just greater than which value:
A. 0.725
B. 0.730
C. 0.720
D. 0.715

Question 6: Use the best mtry and min_n values from Question 5 to finalize the workflow and fit the model to training set. Examine variable importance.

Which variable is most important?
A. Oscore
B. Cscore
C. Impulsive
D. SS

Question 7: To four decimal places, what is the accuracy of your model on the training set?

Question 8: To four decimal places, what is the naive accuracy (training set)?

Question 9: To four decimal places, what is your model's accuracy on the testing set?

Question 10 The difference in accuracy between the training and testing sets implies?

A. Overfitting does not appear to be occurring
B. Overfitting is likely occurring
C. It is not clear whether or not overfitting is occurring