CMM510 Data Mining (RESIT)

Afreen Fatima

2022-07-01

OVERVIEW:

We are using a dataset “swPollution.csv” that shows the records of waste and pollution data. We will analyse and visualise this dataset and perform data mining tasks to discuss, compare and contrast the advantages and disadvantages of applying any data mining techniques that are required to perform the learning task. Further, we will use toolkit to develop a data mining application for a given learning task and evaluating the results obtained. Interpreting the results of tasks by effectively understanding the strengths and limitations of data mining technology and selecting the most appropriate technique to evaluate. To demonstrate knowledge of the state of the art in data mining to understand the current areas in research. To quickly understand and adapt the most appropriate data mining technique to given problems.

Keeping in mind the page limit I have eliminated few codes output lines from this knitted file say from variable importance I took out least important variables, k folds and repeats of algorithms. Although they can be found in RMD file.

```r
# Loading libraries
library(ggplot2)
library(corrplot)

library(ggcorrplot)

library(magrittr)
library(dplyr)

library(grid)
library(tidyverse)

library(caret)

library(mlbench)
library(rpart)
library(C50)
library(partykit)

library(rattle)

library(RColorBrewer)
library(scales)

library(cluster)
library(rgl)

library(pvclust)

library(fpc)
```

Loading dataset:

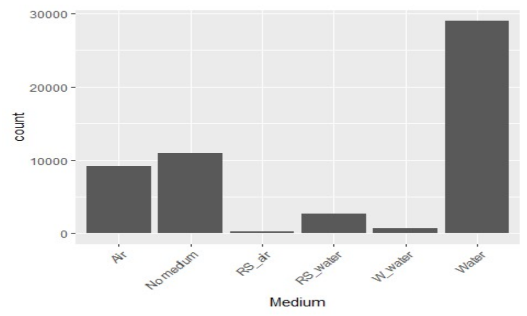
```r
swPollution <- read.csv("swPollution.csv", header=T, stringsAsFactors = T)
```

TASK1: Exploratory Data Analysis:

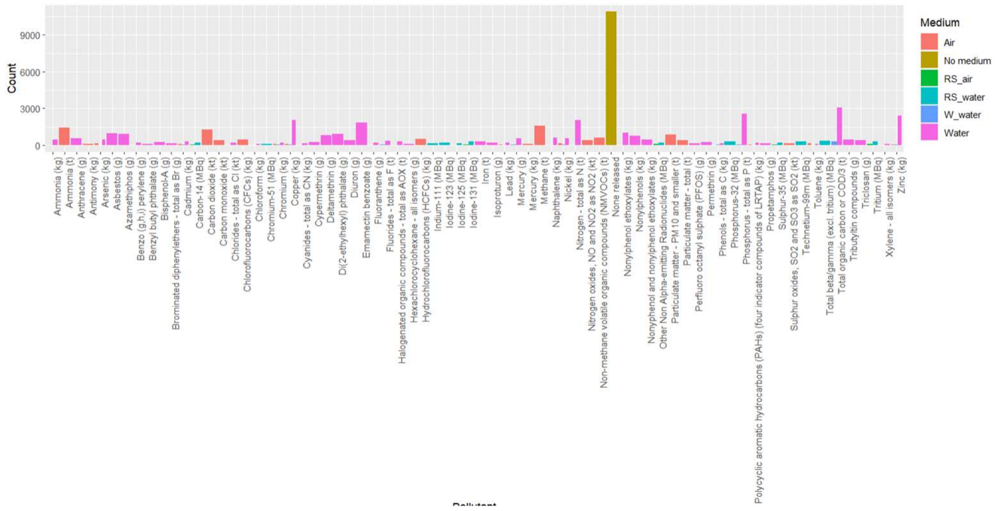
Checking summary of dataset to see descriptive statistics and distribution. Dataset has 9 categorical and 5 numerical variables. No cleaning required and no missing values presen. Only data transformation that is Factor columns has been converted to character except Medium and Integer column Year converted to numeric to have consistency in all numerical attributes. Univariate analysis for all nominal and numerical variables has been done using summary and histograms. Medium has “Water” distribution highest and least in “RS_air”. Facility has highest distribution of “Edinburgh Sewage Treatment Works, Leith” 530. Region has highest distribution of “Glasgow and Strathclyde”. Pollutant is “Total organic carbon or COD/3 (t)” highly distributed. Type has “Pollutant” highly distributed. TransferType has “Disposal” highly distributed. TransferDestination has “Domestic(UK)” highly distriburted. Unit has “T” highly distributed and least is “Kt”. Bivariate analysis between Medium and Pollutant shows “Water” medium has highest distribution with pollutant “Total organic carbon or COD/3 (t)”. “Glassgow and Strathclyde” has highest distribution of medium “Water”. “Waste and waste water management” and “Intensive livestock production and aquaculture” is highly distributed among mediums “Air” and “Water”. “Pollutant” Type is highly distributed with medium “Water”. “Water” Pollutant is highly distributed in unit “g”, “Kg” and “T”

```
summary(swPollution) ## Summarising data

##                                        Facility        Easting
##  Edinburgh Sewage Treatment Works, Leith  :  530   Min.   :     0
##  Dalmuir STW, Beardmore Street, Clydebank :  526   1st Qu.:236740
##  Meadowhead Sewage Treatment Works, Irvine:  493   Median :291675
##  Nigg WWTW, Aberdeen                      :  470   Mean   :288661
##  Hatton STW, By Hatton Farm, Arbroath     :  464   3rd Qu.:339521
##  Shieldhall STW, 38 Renfrew Rd, Glasgow   :  463   Max.   :467497
##  (Other)                                  :49878
##     Northing                                                 Region
##  Min.   : 543423   Glasgow and Strathclyde                      :12623
##  1st Qu.: 666331   No local authority                           :11868
##  Median : 692128   Aberdeen and North East                      : 6277
##  Mean   : 758632   Tayside, Central and Tayside, Central and Fife: 5478
##  3rd Qu.: 829008   Edinburgh and Lothians                       : 4354
##  Max.   :1208300   Tayside, Central and Fife                    : 4183
##                    (Other)                                      : 8041
##                                                          Description
##  Waste and waste-water management                           :26043
##  Intensive livestock production and aquaculture             :14718
##  Energy sector                                              : 3699
##  Radioactive Substances Act Activities                      : 2957
##  Chemical industry                                          : 1517
##  Animal and vegetable products from the food and beverage sector: 1125
##  (Other)                                                    : 2765
##       Year                                Pollutant           Medium
##  Min.   :2007   None released                    :10909   Air      : 9162
##  1st Qu.:2010   Total organic carbon or COD/3 (t): 3368   No medium:10909
##  Median :2014   Phosphorus - total as P (t)      : 2566   RS_air   :  293
##  Mean   :2014   Zinc (kg)                        : 2489   RS_water : 2721
##  3rd Qu.:2017   Copper (kg)                      : 2185   W_water  :  734
##  Max.   :2020   Nitrogen - total as N (t)        : 2078   Water    :29005
##                 (Other)                          :29229
##                   Type         TransferType          TransferDestination
##  Hazardous Waste    : 4561   Disposal: 5722   Domestic (UK)      :10844
##  Non-hazardous Waste: 6348   None    :41917   Export (outside UK):   65
##  Pollutant          :41915   Recovery: 5185   Not applicable     :41915
##
##
##
##
##      Amount           ReportingThreshold   Unit
##  Min.   :        0   Min.   :      0.0   g  : 9931
##  1st Qu.:       17   1st Qu.:      1.0   Kg :12629
##  Median :      117   Median :      5.0   Kt : 2430
##  Mean   :   149881   Mean   :    505.2   MBq: 3014
##  3rd Qu.:      680   3rd Qu.:     50.0   T  :24820
##  Max.   :364000000   Max.   :1000000.0
##
```

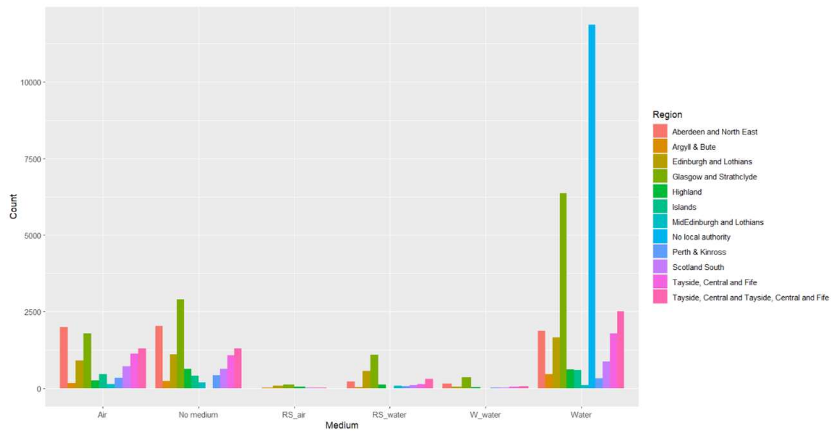
```
dim(swPollution) # Checking dimensions

## [1] 52824    14

## Exploring nominal variables
# Medium bar plot
p <- ggplot(data=swPollution, aes(x=Medium))
p <- p + geom_bar()+  theme(axis.text.x = element_text(angle = 45,
                                  hjust = 1))
p
```



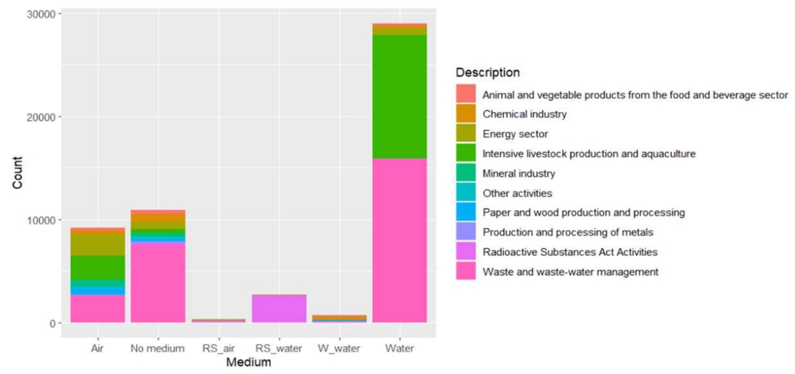
```
## Bivariate analysis with medium
#Grouped bar plot with medium and Pollutant
ggplot(swPollution) +
  geom_bar(aes(x = Pollutant, fill = Medium), position = "dodge") +
  xlab("Pollutant") + ylab("Count")+  theme(axis.text.x = element_text(angle = 90,
                                  hjust = 1))
```



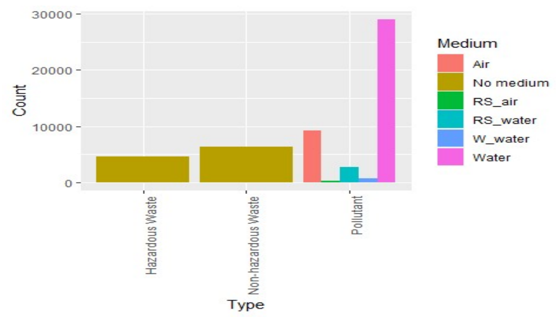
```
#Medium and facility PLOT has lots of noise to visualise better
myggplotF <- ggplot(swPollution) +
  geom_bar(aes(x = Facility, fill = Medium), position = "dodge") +
  xlab("Facility") + ylab("Count")+theme(axis.text.x = element_text(angle = 90,
      hjust = 1))
myggplotF + theme(axis.text = element_text(size=0.1))

#Grouped bar plot with medium and Pollutant
ggplot(swPollution) +
  geom_bar(aes(x = Medium, fill = Region), position = "dodge") +
  xlab("Medium") + ylab("Count")
```



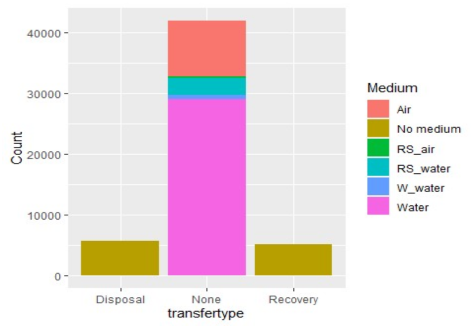
```
# grouped bar plot of DEscription and medium
ggplot(swPollution) +
  geom_bar(aes(x = Medium, fill = Description)) +
  xlab("Medium") + ylab("Count")
```



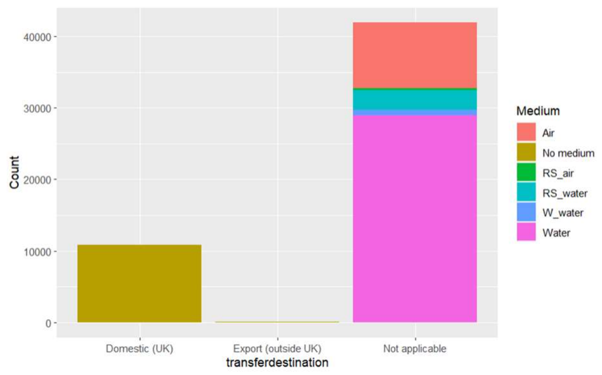
```
#Grouped bar plot with medium and Type
ggplot(swPollution) +
  geom_bar(aes(x = Type, fill = Medium), position = "dodge") +
  xlab("Type") + ylab("Count")+  theme(axis.text.x = element_text(angle = 90,
                                   hjust = 1))
```



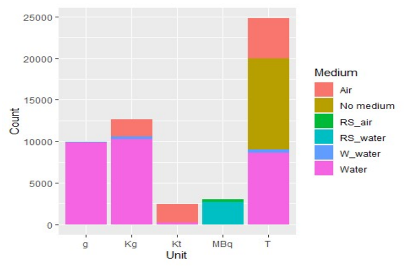
```
# grouped bar plot of transfer type and medium
ggplot(swPollution) +
  geom_bar(aes(x = TransferType, fill = Medium)) +
  xlab("transfertype") + ylab("Count")
```



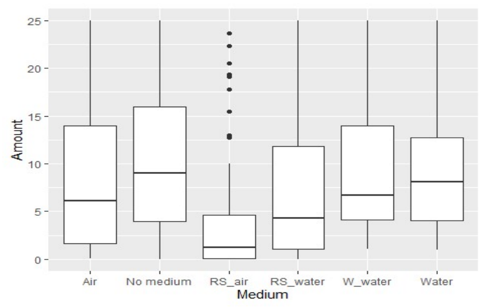
```
# grouped bar plot of transfer destination and medium
ggplot(swPollution) +
  geom_bar(aes(x = TransferDestination, fill = Medium)) +
  xlab("transferdestination") + ylab("Count")
```



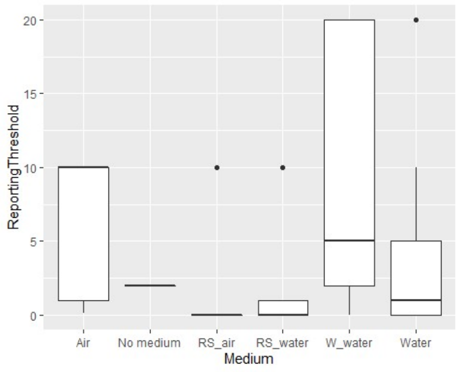
```
# grouped bar plot of Unit and medium
ggplot(swPollution) +
  geom_bar(aes(x = Unit, fill = Medium)) +
  xlab("Unit") + ylab("Count")
```



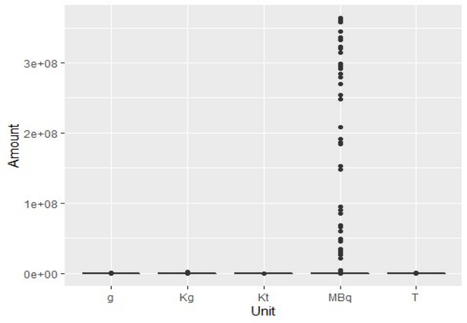
```
# boxplot plot
qplot(Medium, Amount, data = swPollution, geom = "boxplot", ylim = c(0,25))
```



```r
# boxplot plot for medium and reporting threshold
qplot(Medium, ReportingThreshold, data = swPollution, geom = "boxplot", ylim = c(0,20))
```



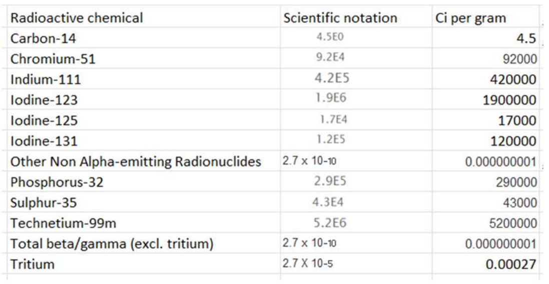
```r
# boxplot plot
qplot(Unit, Amount, data = swPollution, geom = "boxplot")
```



TASK2:

a:

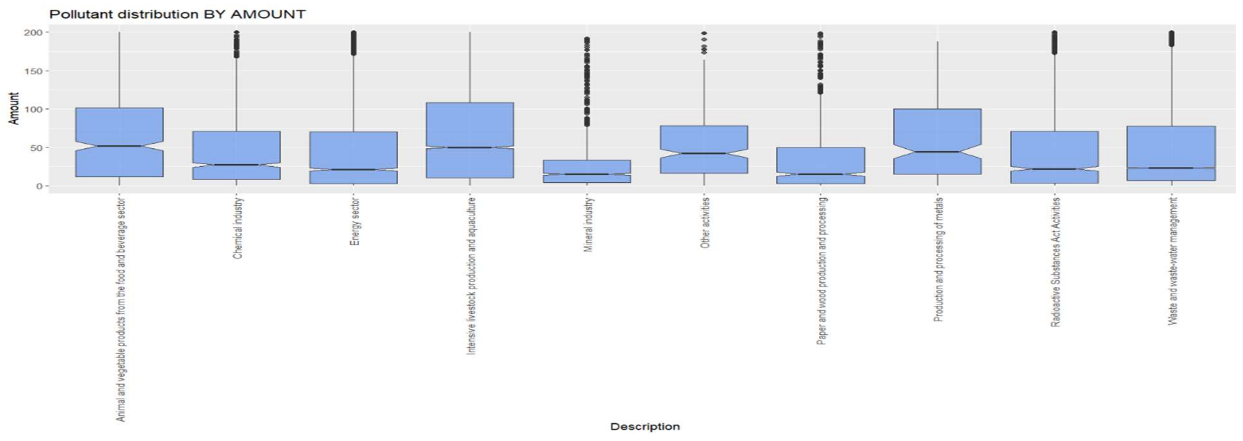
Method1: No conversion of units: Boxplots are not consistent and have different units to be interpreted. So, method 2 shows the conversion of units into tonne Method2: . In order to have consistency and be able analyse and interpret correctly the units have been converted to tonnes. G to t, Kg to t, Kt to t and MBq to Ci Ci to pollutant/per gram then gram to tonnes.



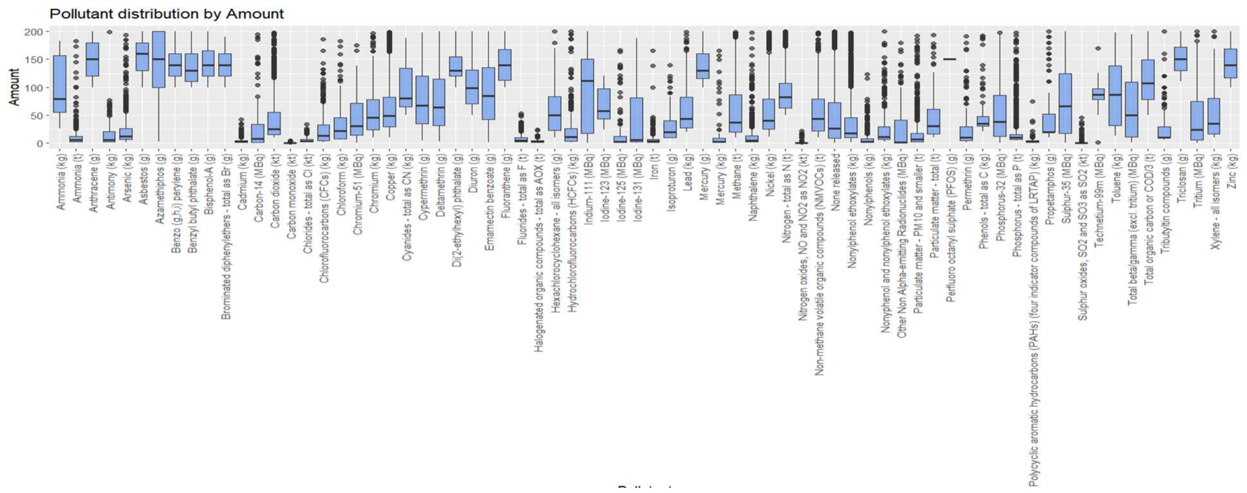
Radioactive chemical	Scientific notation	Ci per gram
Carbon-14	4.5E0	4.5
Chromium-51	9.2E4	92000
Indium-111	4.2E5	420000
Iodine-123	1.9E6	1900000
Iodine-125	1.7E4	17000
Iodine-131	1.2E5	120000
Other Non Alpha-emitting Radionuclides	2.7 x 10-10	0.000000001
Phosphorus-32	2.9E5	290000
Sulphur-35	4.3E4	43000
Technetium-99m	5.2E6	5200000
Total beta/gamma (excl. tritium)	2.7 x 10-10	0.000000001
Tritium	2.7 X 10-5	0.00027

Note: Each radioactivity has it's own scientific notation as per the link provided http://www.iem-inc.com/information/tools/specific-activities that has been calculated after converting MBq to Ci for each radioactive pollutant and converted into grams, that is very very minimal. Say, 1Ci(Curie) = 37000 MBq and there are 4.5E0 (4.5) Curries of Carbon-14 in 1 gram and gram to tonnes has been converted using mutate and ifelse function. The resulting column has been converted to numeric and renaming with proper convention as "AmountInT" has been done. Column year deleted. Also strings from description say units has been cleaned showing only description without units. Description plot can easily be interpreted as Production and processing of metals with highest distribution of pollutants. Pollutants Carbon dioxide and Chlorides-total as CI has the highest distribution among other pollutants in tonnes.

```
# Method1:.Boxplot plot description and amount
ggplot(swPollution, aes(x = Description,
                        y = Amount)) +
  geom_boxplot(notch = TRUE,
               fill = "cornflowerblue",
               alpha = .7) +
  labs(title = "Pollutant distribution BY AMOUNT") + theme(axis.text.x = element_text(angle = 90, vjust =
0.5, hjust=1))+ ylim(0, 200)

## Warning: Removed 21245 rows containing non-finite values (stat_boxplot).
```



```
# boxplot plot of pollutant and amount
ggplot(swPollution, aes(x = Pollutant,
                        y = Amount)) +
  geom_boxplot(notch = FALSE,
               fill = "cornflowerblue",
               alpha = .7) +
  labs(title = "Pollutant distribution by Amount") + theme(axis.text.x = element_text(angle = 90, vjust =
0.5, hjust=1))+ ylim(0, 200)

## Warning: Removed 21245 rows containing non-finite values (stat_boxplot).
```



```
# Method2: converting everything to Tonnes
library(measurements)
# gram to Tonnes
gToT <- swPollution %>%
  mutate(Amount=ifelse(Unit=="g", (Amount)/1000000,Amount))
#Removing g from column concatenated
#gToT$Pollutant<-gsub("[(g)]","",gToT$Pollutant)
# kg to tonne
```

```r
kgToT <- gToT %>%
  mutate(Amount=ifelse(Unit=="Kg", (Amount)/1000,Amount))
# Kilotonne to kg
ktToT <- kgToT %>%
  mutate(Amount=ifelse(Unit=="Kt", (Amount)*1000,Amount))
# Kilotonne to kg
MBqToCi <- ktToT %>%
  mutate(Amount=ifelse(Unit=="MBq", (Amount)/37000,Amount))
#carbon-14 ci to Tonnes
carbon14 <- MBqToCi %>%
  mutate(Amount=ifelse(Pollutant=="Carbon-14 (MBq)", (Amount)/0.0000045,Amount))
#chromium-51 ci to Tonnes
chromium51 <- carbon14 %>%
  mutate(Amount=ifelse(Pollutant=="Chromium-51 (MBq)", (Amount)/0.092,Amount))
#indium-111 ci to Tonnes
indium111 <- chromium51 %>%
  mutate(Amount=ifelse(Pollutant=="Indium-111 (MBq)", (Amount)/0.42,Amount))
#iodine-123 ci to Tonnes
iodine123 <- indium111 %>%
  mutate(Amount=ifelse(Pollutant=="Iodine-123 (MBq)", (Amount)/1.9,Amount))
#iodine-125 ci to Tonnes
iodine125 <- iodine123 %>%
  mutate(Amount=ifelse(Pollutant=="Iodine-125 (MBq)", (Amount)/0.017,Amount))
#iodine-131 ci to Tonnes
iodine131 <- iodine125 %>%
  mutate(Amount=ifelse(Pollutant=="Iodine-131 (MBq)", (Amount)/0.12,Amount))
#phosphorus-32 ci to Tonnes
phosphorus32 <- iodine131 %>%
  mutate(Amount=ifelse(Pollutant=="Phosphorus-32 (MBq)", (Amount)/0.29,Amount))
#sulphur-35 ci to Tonnes
sulphur35 <- phosphorus32 %>%
  mutate(Amount=ifelse(Pollutant=="Sulphur-35 (MBq)", (Amount)/0.043,Amount))
#technetium-99m ci to Tonnes
technetium99m <- sulphur35 %>%
  mutate(Amount=ifelse(Pollutant=="Technetium-99m (MBq)", (Amount)/5.2,Amount))
#tritium ci to Tonnes
tritium <- technetium99m %>%
  mutate(Amount=ifelse(Pollutant=="Tritium (MBq)", (Amount)/0.0097,Amount))
nonalpha <- tritium %>%
  mutate(Amount=ifelse(Pollutant=="Other Non Alpha-emitting Radionuclides (MBq)", (Amount)/0,Amount))
#Total beta/gamma (excl. tritium) ci to Tonnes
swPollutionClean <- nonalpha %>%
  mutate(Amount=ifelse(Pollutant=="Total beta/gamma (excl. tritium) (MBq)", (Amount)/0,Amount))
# Cleaning description column and eliminating unit string of units
swPollutionClean$Pollutant <- as.character(swPollutionClean$Pollutant)
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Ammonia (kg)"] <- "Ammonia"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Ammonia (t)"] <- "Ammonia"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Anthracene (g)"] <- "Anthracene"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Antimony (kg)"] <- "Antimony"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Arsenic (kg)"] <- "Arsenic"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Asbestos (g)"] <- "Asbestos"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Azamethiphos (g)"] <- "Azamethiphos"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Benzo (g,h,i) perylene (g)"] <- "Benzo (g,h,i)
perylene"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Benzyl butyl phthalate (g)"] <- "Benzyl butyl
phthalate"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Bisphenol-A (g)"] <- "Bisphenol-A"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Brominated diphenylethers - total as Br (g)"] <-
"Brominated diphenylethers - total as Br"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Cadmium (kg)"] <- "Cadmium"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Carbon dioxide (kt)"] <- "Carbon dioxide"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Carbon monoxide (kt)"] <- "Carbon monoxide"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Carbon-14 (MBq)"] <- "Carbon-14(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Chlorides - total as Cl (kt)"] <- "Chlorides -
total as Cl"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Chlorofluorocarbons (CFCs) (kg)"] <-
"Chlorofluorocarbons (CFCs"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Chloroform (kg)"] <- "Chloroform"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Chromium (kg)"] <- "Chromium"
```

```
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Chromium-51 (MBq)"] <- "Chromium-
51(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Copper (kg)"] <- "Copper"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Cyanides - total as CN (kg)"] <- "Cyanides -
total as CN"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Cypermethrin (g)"] <- "CypermethrinAmmonia"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Deltamethrin (g)"] <- "Deltamethrin"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Di(2-ethylhexyl) phthalate (g)"] <- "Di(2-
ethylhexyl) phthalate"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Diuron (g)"] <- "Diuron"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Emamectin benzoate (g)"] <- "Emamectin benzoate"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Fluoranthene (g)"] <- "Fluoranthene"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Fluorides - total as F (t)"] <- "Fluorides -
total as F"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Halogenated organic compounds - total as AOX
(t)"] <- "Halogenated organic compounds - total as AOX"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Hexachlorocyclohexane - all isomers (g)"] <-
"Hexachlorocyclohexane - all isomers"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Hydrochlorofluorocarbons (HCFCs) (kg)"] <-
"Hydrochlorofluorocarbons (HCFCs)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Indium-111 (MBq)"] <- "Indium-111(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Iodine-123 (MBq)"] <- "Iodine-123(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Iodine-123 (MBq)"] <- "Iodine-123(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Iodine-131 (MBq)"] <- "Iodine-131(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Iron (t)"] <- "Iron"
swPollutionClean$Pollutant <- as.character(swPollutionClean$Pollutant)
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Isoproturon (g)"] <- "Isoproturon"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Lead (kg)"] <- "Lead"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Mercury (g)"] <- "Mercury"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Mercury (kg)"] <- "Mercury"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Methane (t)"] <- "Methane"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Naphthalene (kg)"] <- "Naphthalene"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Nickel (kg)"] <- "Nickel"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Nitrogen - total as N (t)"] <- "Nitrogen - total
as N"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Nitrogen oxides, NO and NO2 as NO2 (kt)"] <-
"Nitrogen oxides, NO and NO2 as NO2"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Non-methane volatile organic compounds (NMVOCs)
(t)"] <- "Non-methane volatile organic compounds (NMVOCs)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Nonylphenol ethoxylates (kg)"] <- "Nonylphenol
ethoxylates"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Nonylphenols (kg)"] <- "Nonylphenols"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Nonyphenol and nonylphenol ethoxylates (kg)"] <-
"Nonyphenol and nonylphenol ethoxylates"
#DEleting instances of other non alpha and total beta gamma as they are total of the radionuclides already
interpreted.
swPollutionClean <- swPollutionClean[swPollutionClean$Pollutant!="Total beta/gamma (excl. tritium)
(MBq)",]
swPollutionClean <- swPollutionClean[swPollutionClean$Pollutant!="Other Non Alpha-emitting Radionuclides
(MBq)",]
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Particulate matter - PM10 and smaller (t)"] <-
"Particulate matter - PM10 and smaller"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Particulate matter - total (t)"] <- "Particulate
matter - total"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Perfluoro octanyl sulphate (PFOS) (g)"] <-
"Perfluoro octanyl sulphate (PFOS)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Permethrin (g)"] <- "Permethrin"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Phenols - total as C (kg)"] <- "Phenols - total
as C"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Phosphorus - total as P (t)"] <- "Phosphorus -
total as P"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Phosphorus-32 (MBq)"] <- "Phosphorus-
32(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Polycyclic aromatic hydrocarbons (PAHs) (four
indicator compounds of LRTAP) (kg)"] <- "Polycyclic aromatic hydrocarbons (PAHs) (four indicator compounds
of LRTAP)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Propetamphos (g)"] <- "Propetamphos"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Sulphur oxides, SO2 and SO3 as SO2 (kt)"] <-
"Sulphur oxides, SO2 and SO3 as SO2"
```

```
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Sulphur-35 (MBq)"] <- "Sulphur-35(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Technetium-99m (MBq)"] <- "Technetium-
99m(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Toluene (kg)"] <- "Toluene"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Total organic carbon or COD/3 (t)"] <- "Total
organic carbon or COD/3"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Tributyltin compounds (g)"] <- "Tributyltin
compounds"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Triclosan (g)"] <- "Triclosan"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Tritium (MBq)"] <- "Tritium(radioactive)"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Xylene - all isomers (kg)"] <- "Xylene - all
isomers"
swPollutionClean$Pollutant[swPollutionClean$Pollutant == "Zinc (kg)"] <- "Zinc"
#Removing column Year
swPollutionClean$Year <-NULL
#swPollutionClean$Unit <- NULL
# Too many values to run an algorithms causing delay and failing predictions
swPollutionClean$Facility <- NULL
colnames(swPollutionClean)[colnames(swPollutionClean) == "Amount"] <- "AmountInT" ## Renaming column name
```

b:

Options for data preparation if there are 50 missing values in TransferType column: Summarising data of TransferType in task1, none has larger distribution than Recovery and Disposal. Deleting 5o instances from the least distributed may make it more less or so. But not noticable. • Delete instances with missing values as we have 52196 total instances in cleaned dataset and it's not going to affect the learning with removal of 50 instances. • Assigning a value to the instances with the missing value with mean or median value • Process of data cleaning where we identify incorrect noise, saying or interpreting this is not missing values but wrong values, example, Some human is 30 meter tall that's wrong. (Reference: lecture 10 data mining)

c:

Using clean dataset swPollutionClean derived from Task2a 10% of data partition has been done with class Medium. Then converted integer list "swpTen" to dataframe that consist of 10% of dataset "swPollution" consisting of instances 5223. Summarised to ensure no levels has been dropped. Using tip 1: From "swpTen" further "partition" function has been created to create data partition of 50% test and 50% train that will constitute 5% of train data and 5% of test data from swPollution. Ensured no levels has been dropped by converting character columns to factor and summarising. swpTrain has 2613 instances and swpTest has 2610 instances. Although they are not equal by 3 instances but I have ensured in code that initially 10% from main dataset swPollution has been taken with a stratified split and swpTen having 5223 instances. From swpTen 0.50 partition has been created for train set that is initially allocating 2613 instances to train data and left over are 2610 instances.

```
#Producing 10% of swPollutionClean
set.seed(2) #Ensuring reproducibility
swpTen <- createDataPartition(y = swPollutionClean$Medium,
                              p = 0.10,
                              list = FALSE)
swpTen <- swPollutionClean[swpTen,]#Putting the resulting dataset into dataframe

# Further producing two datasets from swpTrain and swpTest that is 50-50 of swpTen
set.seed(2)
partition <- createDataPartition(y = swpTen$Medium, p = 0.50,list = FALSE)
swpTrain <- swpTen[partition,]
swpTest  <- swpTen[-partition,]
summary(swpTrain) ##ensure distribution
```

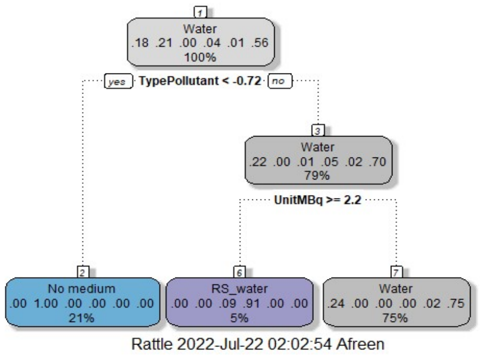
Task3:

The variables in swpTrain and swpTest has been converted from character to factor except the "Facility" to character. Experiment design for CART (rpart), ctree and K nearest neighbour has been done. Where for each experiment a seed as been set to ensure reproducibility. Training method is used for train control where method is "repeatedcv" that is repeated cross validation that will have 8 folds and repeat it for 3 times. In 3 of the experiments medium as a class data used is "swpTrain", method is "rpart" , "c5tree" and "knn". rpart and c5 preprocessing using center and scaling is done as the values in numeric columns are not standardised and not in scale, metric is "Accuracy" . In confusion matrix predictions are in columns and Medium values are in rows. For CART with complexity parameter 0.08 accuracy is 0.8294 and kappa is 0.6902. Tree classifier with C5.0Tree accuracy is 0.72 with kappa 0.66. For instance based classifier Knn tuneGrid set to expand.grid k value as odd values 1,3,5,7 with train control repeated cross validation of 8 folds and 3 repeats. Accuracy of 0.96 with kappa 0.94 and k=1. Comparative analysis of 3 experiments shows Knn has highest accuracy and kappa, whereas C5.0 is ranging accuracy from 0.6 – 0.85 on scales. Note: Algorithms was taking time to load and run then had a look on tip and deleted the attribute Facility as it was giving lots of errors with it in output code and predictions was failing. Revisited task2 and deleted Facility.

swpTrainExperiment	Accuracy	Kappa	samples	metric	High Attribute usage
CART	0.82	0.69	2613	Accuracy, preProcess	• TypePollutant • TransferDestination NotApplicable • PollutantNoneReleased
C5TREE	0.55 Size = 34, Error = 2%	0.44	2613	Accuracy, preProcess tuneLenght	• Pollutant none released, • UnitMBq
KNN	0.84	0.73	2613	Accuracy, tuneGrid	• Type • Transfer Destination

```
# Preparing training method, verboseIter was set to false initially and true next to avoid large amounts
of output
ctrl <- trainControl(method="repeatedcv", number=8, repeats=3, verboseIter=TRUE)
#CART
set.seed(3)
mod.cart<- train(Medium~., data = swpTrain, method="rpart", preProcess=c("center","scale"),
                 metric="Accuracy", trControl=ctrl)

mod.cart

## CART
##
## 2613 samples
##   11 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## Pre-processing: centered (101), scaled (101)
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2287, 2285, 2287, 2288, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.08433735  0.8294583  0.6902983
##   0.09380379  0.7889305  0.5991845
##   0.46987952  0.6678927  0.2929082
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.08433735.

varImp(mod.cart)

## rpart variable importance
##
##   only 20 most important variables shown (out of 106)
##
##                                                               Overall
## TypePollutant                                                 100.00
## TransferDestinationNot applicable                             100.00
## PollutantNone released                                        100.00
## TransferTypeNone                                               99.77
## TypeNon-hazardous Waste                                        53.40
## UnitMBq                                                        24.47
## DescriptionRadioactive Substances Act Activities               23.84
```

```r
#summary(mod.cart)
fancyRpartPlot(mod.cart$finalModel)
```



```r
confusionMatrix(mod.cart, norm = "none")

## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##              Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air        219         0      0        0       7    35
##   No medium    0      1638      0        0       0     0
##   RS_air       0         0      0        0       0     0
##   RS_water     0         0     33      327       0     0
##   W_water      0         0      0        0       0     0
##   Water     1158         0      0        0     104  4318
##
##  Accuracy (average) : 0.8294

#ctree
set.seed(3)
mod.c5tree <- train(Medium ~ ., data = swpTrain,
             method = "C5.0Tree",
             metric = "Accuracy",
             preProcess=c("center","scale"),
             tuneLength = 12, #For caret to select default 5 settings to tune
             trControl = ctrl)mod.c5tree

## Single C5.0 Tree
##
## 2613 samples
##   11 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## Pre-processing: centered (101), scaled (101)
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2287, 2285, 2287, 2288, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.5564508  0.4568156

#Checking variable importance
varImp(mod.c5tree)

## C5.0Tree variable importance
##
summary(mod.c5tree)

##
## Call:
```

```
## C50:::C5.0.default(x = x, y = y, weights = wts)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri Jul 22 02:03:31 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 2613 cases (102 attributes) from undefined.data
##
## Decision tree:
##
## PollutantNone released > -0.5138578: No medium (546)
## PollutantNone released <= -0.5138578:
## :...UnitMBq > -0.2193544:
##     :...DescriptionEnergy sector <= -0.2743717: RS_water (112/6)
##     :   DescriptionEnergy sector > -0.2743717: RS_air (8/3)
##     UnitMBq <= -0.2193544:
##     :...PollutantMethane > -0.1765335: Air (79)
##         PollutantMethane <= -0.1765335:
##         :...PollutantCarbon dioxide > -0.1424668: Air (52)
##             PollutantCarbon dioxide <= -0.1424668:
##             :...PollutantParticulate matter - PM10 and smaller > -0.1338391: Air (46)
##                 PollutantParticulate matter - PM10 and smaller <= -0.1338391: [S1]
##
## SubTree [S1]
##
## PollutantNon-methane volatile organic compounds (NMVOCs) > -0.118171: Air (36)
## PollutantNon-methane volatile organic compounds (NMVOCs) <= -0.118171:
## :...PollutantCarbon monoxide > -0.1077495: Air (30)
##     PollutantCarbon monoxide <= -0.1077495:
##     :...PollutantHydrochlorofluorocarbons (HCFCs) > -0.1002318: Air (26)
##         PollutantHydrochlorofluorocarbons (HCFCs) <= -0.1002318:
##         :...DescriptionPaper and wood production and processing > -0.1308462:
##             :...PollutantTotal organic carbon or COD/3 <= -0.2645709: Air (22/4)
##             :   PollutantTotal organic carbon or COD/3 > -0.2645709: W_water (3/1)
##             DescriptionPaper and wood production and processing <= -0.1308462:
##             :...DescriptionChemical industry > -0.1694968:
##                 :...PollutantNitrogen oxides, NO and NO2 as NO2 > -0.08999306: Air (2)
##                 :   PollutantNitrogen oxides, NO and NO2 as NO2 <= -0.08999306:
##                 :   :...RegionEdinburgh and Lothians > -0.3016424: Air (8/2)
##                 :       RegionEdinburgh and Lothians <= -0.3016424:
##                 :       :...ReportingThreshold > -0.03263058: Air (3/1)
##                 :           ReportingThreshold <= -0.03263058: [S2]
##                 DescriptionChemical industry <= -0.1694968:
##                 :...UnitKt > -0.2174316: [S3]
##                     UnitKt <= -0.2174316:
##                     :...RegionNo local authority > -0.546437: Water (601)
##                         RegionNo local authority <= -0.546437: [S4]
##
## SubTree [S2]
##
## RegionGlasgow and Strathclyde > -0.5552879: W_water (7/1)
## RegionGlasgow and Strathclyde <= -0.5552879:
## :...AmountInT <= -0.04861555: Water (7/1)
##     AmountInT > -0.04861555: W_water (4/1)
##
## SubTree [S3]
##
## DescriptionWaste and waste-water management <= -0.9956085: Air (20)
## DescriptionWaste and waste-water management > -0.9956085: Water (10)
##
## SubTree [S4]
##
## DescriptionIntensive livestock production and aquaculture > -0.6290325:
## :...Easting <= -1.333757: Water (5)
## :   Easting > -1.333757: Air (72)
## DescriptionIntensive livestock production and aquaculture <= -0.6290325:
## :...PollutantChlorofluorocarbons (CFCs > -0.09212861: Air (22)
##     PollutantChlorofluorocarbons (CFCs <= -0.09212861:
##     :...DescriptionWaste and waste-water management > -0.9956085: Water (794/22)
##         DescriptionWaste and waste-water management <= -0.9956085:
##         :...PollutantTotal organic carbon or COD/3 > -0.2645709:
##             :...AmountInT <= -0.04160006: W_water (11/1)
##             :   AmountInT > -0.04160006: Water (2)
##             PollutantTotal organic carbon or COD/3 <= -0.2645709:
##             :...DescriptionMineral industry > -0.1410628: Air (8)
##                 DescriptionMineral industry <= -0.1410628:
##                 :...PollutantAntimony > -0.04796473: Air (4)
##                     PollutantAntimony <= -0.04796473:
##                     :...PollutantParticulate matter - total > -0.08999306: Air (4)
##                         PollutantParticulate matter - total <= -0.08999306: [S5]
##
## SubTree [S5]
##
## PollutantPolycyclic aromatic hydrocarbons (PAHs) (four indicator compounds of LRTAP) > -0.07338001: Air
(2)
## PollutantPolycyclic aromatic hydrocarbons (PAHs) (four indicator compounds of LRTAP) <= -0.07338001:
## :...DescriptionOther activities > -0.08999306: Air (2/1)
##     DescriptionOther activities <= -0.08999306:
##     :...PollutantNickel <= -0.1308462: Water (57/9)
##         PollutantNickel > -0.1308462:
##         :...ReportingThreshold <= -0.0346645: Air (6)
##             ReportingThreshold > -0.0346645: Water (2)
##
##
## Evaluation on training data (2613 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      34   53( 2.0%)   <<
##
##
##     (a)   (b)   (c)   (d)   (e)   (f)    <-classified as
##    ----  ----  ----  ----  ----  ----
##     436                       1    22    (a): class Air
##           546                            (b): class No medium
##                   5     6                (c): class RS_air
##                   3   106                (d): class RS_water
##       6                      21    10    (e): class W_water
##       2                       3  1446    (f): class Water
##
##
##  Attribute usage:
##
##  100.00% PollutantNone released
##   79.10% UnitMBq
##   74.51% PollutantMethane
##   71.49% PollutantCarbon dioxide
##   69.50% PollutantParticulate matter - PM10 and smaller
##   67.74% PollutantNon-methane volatile organic compounds (NMVOCs)
##   66.36% PollutantCarbon monoxide
##   65.21% PollutantHydrochlorofluorocarbons (HCFCs)
##   64.22% DescriptionPaper and wood production and processing
##   63.26% DescriptionChemical industry
##   62.07% UnitKt
##   60.93% RegionNo local authority
##
##
## Time: 0.3 secs
```

```
print(mod.c5tree$finalModel)
```

```
##
## Call:
## C50:::C5.0.default(x = x, y = y, weights = wts)
```

```
##
## Classification Tree
## Number of samples: 2613
## Number of predictors: 101
##
## Tree size: 34
##
## Non-standard options: attempt to group attributes
```

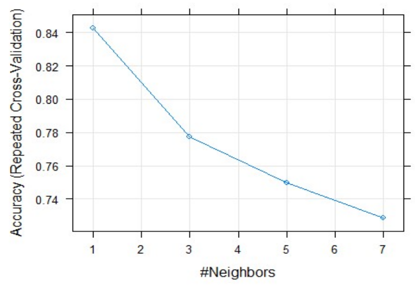
```
confusionMatrix(mod.c5tree, norm = "none")
```

```
## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air       1357         0      0        0      89  3314
##   No medium    0      1638      0        0       0     0
##   RS_air       0         0      6       14       0     0
##   RS_water     0         0     27      313       0     0
##   W_water      3         0      0        0      15     7
##   Water       17         0      0        0       7  1032
##
##  Accuracy (average) : 0.5563
```

```
#Instance based classifier (k-nn)
set.seed(3)
mod.knn<- train(Medium~., data=swpTrain,
        method="knn", tuneGrid=expand.grid(k=c(1,3,5,7)),
        #preProcess=c("center","scale"),
        trControl=ctrl, metric="Accuracy")
```

```
## Aggregating results
## Selecting tuning parameters
## Fitting k = 1 on full training set
```

```
plot(mod.knn)
```



```
mod.knn
```

```
## k-Nearest Neighbors
##
## 2613 samples
##   11 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2287, 2285, 2287, 2288, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   1  0.8426156  0.7395923
##   3  0.7775459  0.6203531
##   5  0.7499797  0.5635788
##   7  0.7286923  0.5153779
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
#checking summary results
summary(mod.knn$finalModel)
```

```
##             Length Class      Mode
## learn         2    -none-     list
## k             1    -none-     numeric
## theDots       0    -none-     list
## xNames      101    -none-     character
## problemType   1    -none-     character
## tuneValue     1    data.frame list
## obsLevels     6    -none-     character
## param         0    -none-     list
```

```
#confusion matrix
confusionMatrix(mod.knn, norm="none")
```

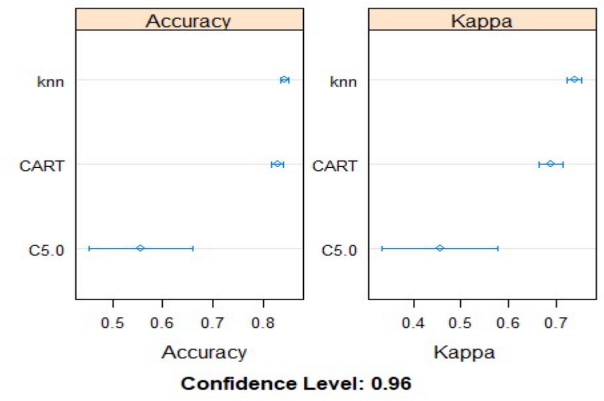
```
## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air       1009       318      2       12      28    85
##   No medium  225      1038     11       24      27    33
##   RS_air       0         1     15        7       0     0
##   RS_water    15        35      4      274       0     3
##   W_water     13        34      0        3      39     2
##   Water      115       212      1        7      17  4230
##
##  Accuracy (average) : 0.8426
```

```
# collect resamples (results for comparison)
results <- resamples(list(CART=mod.cart, C5.0 = mod.c5tree, knn= mod.knn))
results # show accuracy and kappa details
```

```
##
## Call:
## resamples.default(x = list(CART = mod.cart, C5.0 = mod.c5tree, knn = mod.knn))
##
## Models: CART, C5.0, knn
## Number of resamples: 24
## Performance metrics: Accuracy, Kappa
## Time estimates for: everything, final model fit
```

```
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: CART, C5.0, knn
## Number of resamples: 24
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## CART 0.8018293 0.8071917 0.8287462 0.8294583 0.8385631 0.9141104    0
## C5.0 0.4176829 0.4244067 0.4276923 0.5564508 0.5316453 0.9815951    0
## knn  0.8103976 0.8311773 0.8442744 0.8426156 0.8535276 0.8926380    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## CART 0.6335196 0.6424737 0.6913278 0.6902983 0.7080429 0.8575397    0
## C5.0 0.2929519 0.3013688 0.3046184 0.4568156 0.4172130 0.9698734    0
## knn  0.6857329 0.7168905 0.7443290 0.7395923 0.7594439 0.8239578    0

# Showing results graphically
scales <- list(x=list(relation="free"), y=list(relation= "free"))
dotplot(results, scales=scales, conf.level = 0.96)
```



TASK4:

Ensuring the test dataset “swpTest” is processed same way as “swpTrain” that has been used to train the models in task 3 by making datatypes consistent. Testing models built in task 3 for CART C5.0Tree and K-nn using swpTest with parameter “raw” to use all the data from test set “swpTest”. CART has reduced accuracy from train model in overall statistics as 0.80. C5tree increased accuracy to 0.97. While Knn is same as train model with accuracy 0.96. C5Tree and Knn

performed well. While, Knn did well with train and test sets and C5 tree has improved in test accuracy.

swpTestExperiment	Accuracy	Kappa	samples	pvalue	CI
CART	0.80	0.64	2610	0	95%
C5TREE	0.97	0.95	2610	0	95%
KNN	0.86	0.78	2610	0	95%

```
#Testing the models mod.cart, mod.ctree, mod.knn by running them on swpTest and evaluating the results
obtained.
# mod.cart
print(mod.cart)

## CART
##
## 2613 samples
##   11 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## Pre-processing: centered (101), scaled (101)
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2287, 2285, 2287, 2288, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.08433735  0.8294583  0.6902983
##   0.09380379  0.7889305  0.5991845
##   0.46987952  0.6678927  0.2929082
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.08433735.
```

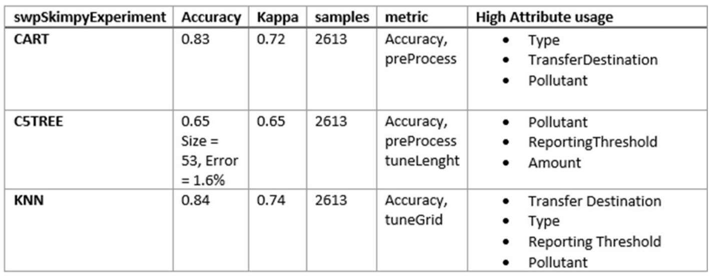
```
#testing rpart on swpTest data
TestCart <- predict(mod.cart, newdata = swpTest, type="raw")
#Printing confusion matrix
confusionMatrix(TestCart, swpTest$Medium)

## Confusion Matrix and Statistics
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air          0         0      0        0       0     0
##   No medium    0       545      0        0       0     0
##   RS_air       0         0      0        0       0     0
##   RS_water     0         0     11      109       0     0
##   W_water      0         0      0        0       0     0
##   Water      458         0      0        0      37  1450
##
## Overall Statistics
##
##                Accuracy : 0.8061
##                  95% CI : (0.7904, 0.8211)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6413
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Air Class: No medium Class: RS_air Class: RS_water
## Sensitivity              0.0000           1.0000      0.000000         1.00000
## Specificity              1.0000           1.0000      1.000000         0.99560
## Pos Pred Value              NaN           1.0000           NaN         0.90833
## Neg Pred Value           0.8245           1.0000      0.995785         1.00000
## Prevalence               0.1755           0.2088      0.004215         0.04176
## Detection Rate           0.0000           0.2088      0.000000         0.04176
## Detection Prevalence     0.0000           0.2088      0.000000         0.04598
## Balanced Accuracy        0.5000           1.0000      0.500000         0.99780
##                      Class: W_water Class: Water
## Sensitivity                 0.00000       1.0000
## Specificity                 1.00000       0.5733
## Pos Pred Value                  NaN       0.7455
## Neg Pred Value              0.98582       1.0000
## Prevalence                  0.01418       0.5556
## Detection Rate              0.00000       0.5556
## Detection Prevalence        0.00000       0.7452
## Balanced Accuracy           0.50000       0.7866

#ctree
print(mod.c5tree)

## Single C5.0 Tree
##
## 2613 samples
##   11 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## Pre-processing: centered (101), scaled (101)
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2287, 2285, 2287, 2288, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.5564508  0.4568156

#testing ctree on swpTest data
TestC5tree <- predict(mod.c5tree, newdata = swpTest, type="raw")
```

```
#Printing confusion matrix
confusionMatrix(TestC5tree, swpTest$Medium)

## Confusion Matrix and Statistics
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air        435         0      0        0       8     3
##   No medium    0       545      0        0       0     0
##   RS_air       0         0      4        5       0     0
##   RS_water     0         0      7      104       0     0
##   W_water      0         0      0        0      19     9
##   Water       23         0      0        0      10  1438
##
## Overall Statistics
##
##                Accuracy : 0.9751
##                  95% CI : (0.9684, 0.9807)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9593
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Air Class: No medium Class: RS_air Class: RS_water
## Sensitivity              0.9498           1.0000      0.363636         0.95413
## Specificity              0.9949           1.0000      0.998076         0.99720
## Pos Pred Value           0.9753           1.0000      0.444444         0.93694
## Neg Pred Value           0.9894           1.0000      0.997309         0.99800
## Prevalence               0.1755           0.2088      0.004215         0.04176
## Detection Rate           0.1667           0.2088      0.001533         0.03985
## Detection Prevalence     0.1709           0.2088      0.003448         0.04253
## Balanced Accuracy        0.9723           1.0000      0.680856         0.97566
##                      Class: W_water Class: Water
## Sensitivity                 0.51351       0.9917
## Specificity                 0.99650       0.9716
## Pos Pred Value              0.67857       0.9776
## Neg Pred Value              0.99303       0.9895
## Prevalence                  0.01418       0.5556
## Detection Rate              0.00728       0.5510
## Detection Prevalence        0.01073       0.5636
## Balanced Accuracy           0.75501       0.9816

# knn
print(mod.knn)

## k-Nearest Neighbors
##
## 2613 samples
##   11 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2287, 2285, 2287, 2288, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   1  0.8426156  0.7395923
##   3  0.7775459  0.6203531
##   5  0.7499797  0.5635788
##   7  0.7286923  0.5153779
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.
```

```
#testing rpart on swpTest data
Testknn <- predict(mod.knn, newdata = swpTest, type="raw")
#Printing confusion matrix
confusionMatrix(Testknn, swpTest$Medium)

## Confusion Matrix and Statistics
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air        351       100      1        2      11    24
##   No medium   60       390      5        7       7    11
##   RS_air       2         0      2        0       0     1
##   RS_water     0        10      3       98       0     0
##   W_water      6         7      0        0      16     1
##   Water       39        38      0        2       3  1413
##
## Overall Statistics
##
##                Accuracy : 0.8697
##                  95% CI : (0.8562, 0.8824)
##     No Information Rate : 0.5556
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7859
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Air Class: No medium Class: RS_air Class: RS_water
## Sensitivity              0.7664           0.7156     0.1818182         0.89908
## Specificity              0.9359           0.9564     0.9988457         0.99480
## Pos Pred Value           0.7178           0.8125     0.4000000         0.88288
## Neg Pred Value           0.9496           0.9272     0.9965451         0.99560
## Prevalence               0.1755           0.2088     0.0042146         0.04176
## Detection Rate           0.1345           0.1494     0.0007663         0.03755
## Detection Prevalence     0.1874           0.1839     0.0019157         0.04253
## Balanced Accuracy        0.8511           0.8360     0.5903319         0.94694
##                      Class: W_water Class: Water
## Sensitivity                 0.43243       0.9745
## Specificity                 0.99456       0.9293
## Pos Pred Value              0.53333       0.9452
## Neg Pred Value              0.99186       0.9668
## Prevalence                  0.01418       0.5556
## Detection Rate              0.00613       0.5414
## Detection Prevalence        0.01149       0.5728
## Balanced Accuracy           0.71350       0.9519
```

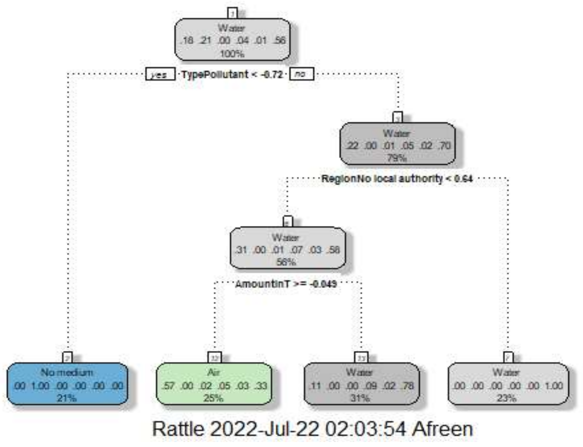
TASK5:

A: Although there has been conversion took place in initial tasks where “Amount” column has been converted to Tonnes from mixed units without concatenation. My column “AmountInT” represents amounts in Tonnes. Deleted Description and Unit column from swpTrain and named the new dataset as swpSkimpy. It has 2613 instances and 10 variables. B: Setting seed to ensure reproducibility. Running experiment CART, C5.0Tree and K-nn on swpSkimpy dataset produced in taskA. CART has increased accuracy and kappa with reduced dataset with metrics Accuracy and preProcess. While, the important variable is Type, Transfer destination and Pollutant. C5Tree has increased accuracy and Kappa with reduced dataset with metrics accuracy and preprocess. Also less error than swpTrain 1.6%. Important variable is Pollutant, Reporting Threshold and Amount. Knn has the same result with accuracy and kappa while their has been change in important variables due to removal of attributes unit and destination. Although Removal of attribute did not deteriorated the model performance, but there is a risk of information loss that we need to be mindful off with the evidence at the time of exploratory data analysis. As Unit was second important attribute at mod.c5tree that node has been terminated.



swpSkimpyExperiment	Accuracy	Kappa	samples	metric	High Attribute usage
CART	0.83	0.72	2613	Accuracy, preProcess	• Type • TransferDestination • Pollutant
C5TREE	0.65 Size = 53, Error = 1.6%	0.65	2613	Accuracy, preProcess tuneLenght	• Pollutant • ReportingThreshold • Amount
KNN	0.84	0.74	2613	Accuracy, tuneGrid	• Transfer Destination • Type • Reporting Threshold • Pollutant

```
## a: Creating swpSkimpy from swpTrain
## Subsetting swpTrain excluding Description and unit to create swpSkimpy
swpSkimpy = subset(swpTrain, select = -c(Description, Unit))

## b: Running CART, C5.0Tree and K-nn on swpSkimpy
##CART with traincontrol ctrl defined with 8 folds of repeated cv and 3 repeats
set.seed(5)
swpSkimpy.cart<- train(Medium~., data = swpSkimpy, method="rpart", preProcess=c("center","scale"),
                 metric="Accuracy", trControl=ctrl)


## Aggregating results
## Selecting tuning parameters
## Fitting cp = 0.062 on full training set

swpSkimpy.cart

## CART
##
## 2613 samples
##    9 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## Pre-processing: centered (88), scaled (88)
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2288, 2285, 2288, 2284, 2287, 2285, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.06196213  0.8326267  0.7220255
##   0.06755594  0.7860558  0.6119765
##   0.46987952  0.6768634  0.3162014
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.06196213.

varImp(swpSkimpy.cart)

## rpart variable importance
##
##   only 20 most important variables shown (out of 94)
##
##                                              Overall
## TypePollutant                                100.000
## TransferDestinationNot applicable            100.000
## PollutantNone released                       100.000
## TransferTypeNone                              99.769
## TypeNon-hazardous Waste                       53.397
## AmountInT                                     36.838
## PollutantMethane                              23.739

#summary(swpSkimpy.cart)
fancyRpartPlot(swpSkimpy.cart$finalModel)
```



```
confusionMatrix(swpSkimpy.cart, norm = "none")

## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##             Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air       1063         0     28       87      38   527
##   No medium    0      1638      0        0       0     0
##   RS_air       0         0      0        0       0     0
##   RS_water     0         0      0        0       0     0
##   W_water      0         0      0        0       0     0
##   Water      314         0      5      240      73  3826
##
##  Accuracy (average) : 0.8326

## C5.0Tree with traincontrol ctrl defined with 8 folds of repeated cv and 3 repeats
set.seed(5)
swpSkimpy.c5tree <- train(Medium ~ ., data = swpSkimpy,
             method = "C5.0Tree",
             metric = "Accuracy",
             preProcess=c("center","scale"),
             tuneLength = 12, #For caret to select default 5 settings to tune
             trControl = ctrl)


## Aggregating results
## Fitting final model on full training set

swpSkimpy.c5tree

## Single C5.0 Tree
##
## 2613 samples
##    9 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## Pre-processing: centered (88), scaled (88)
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2288, 2285, 2288, 2284, 2287, 2285, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6536394  0.5620336

#Checking variable importance
varImp(swpSkimpy.c5tree)
```

```
## C5.0Tree variable importance
##
##   only 20 most important variables shown (out of 103)
##
##                                                       Overall
## PollutantNonereleased                                 100.00
## PollutantMethane                                       79.10
## PollutantCarbondioxide                                 76.08
## PollutantParticulatematter-PM10andsmaller              74.09
## PollutantNon-methanevolatileorganiccompounds(NMVOCs)   72.33
## PollutantHydrochlorofluorocarbons(HCFCs)               70.95
## RegionNolocalauthority                                 69.96
## PollutantPhosphorus-32(radioactive)                    46.96
```

```
summary(swpSkimpy.c5tree)
```

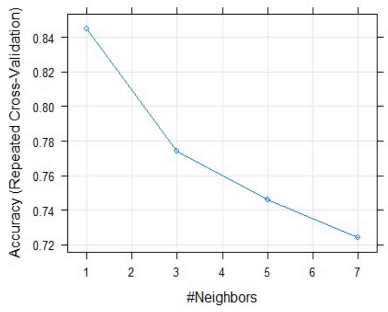
```
##
## Call:
## C50:::C5.0.default(x = x, y = y, weights = wts)
##
##
## C5.0 [Release 2.07 GPL Edition]      Fri Jul 22 02:04:21 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 2613 cases (89 attributes) from undefined.data
##
## Decision tree:
##
## PollutantNone released > -0.5138578: No medium (546)
## PollutantNone released <= -0.5138578:
## :...PollutantMethane > -0.1765335: Air (79)
##     PollutantMethane <= -0.1765335:
##     :...PollutantCarbon dioxide > -0.1424668: Air (52)
##         PollutantCarbon dioxide <= -0.1424668:
##         :...PollutantParticulate matter - PM10 and smaller > -0.1338391: Air (46)
##             PollutantParticulate matter - PM10 and smaller <= -0.1338391: [S1]
##
## SubTree [S1]
##
## PollutantNon-methane volatile organic compounds (NMVOCs) > -0.118171: Air (36)
## PollutantNon-methane volatile organic compounds (NMVOCs) <= -0.118171:
## :...PollutantHydrochlorofluorocarbons (HCFCs) > -0.1002318: Air (26)
##     PollutantHydrochlorofluorocarbons (HCFCs) <= -0.1002318:
##     :...RegionNo local authority > -0.546437: Water (601)
##         RegionNo local authority <= -0.546437:
##         :...PollutantPhosphorus-32(radioactive) > -0.09421736: RS_water (23)
##             PollutantPhosphorus-32(radioactive) <= -0.09421736:
##             :...PollutantTritium(radioactive) > -0.0878073:
##                 :...AmountInT <= -0.04862583: RS_water (12)
##                 :   AmountInT > -0.04862583:
##                 :   :...RegionHighland > -0.1776823: RS_water (2)
##                 :       RegionHighland <= -0.1776823:
##                 :       :...AmountInT <= 1.196549: RS_air (4)
##                 :           AmountInT > 1.196549: RS_water (2)
##                 PollutantTritium(radioactive) <= -0.0878073:
##                 :...PollutantChlorofluorocarbons (CFCs > -0.09212861: Air (22)
##                     PollutantChlorofluorocarbons (CFCs <= -0.09212861:
##                     :...PollutantParticulate matter - total > -0.08999306: Air (21)
##                         PollutantParticulate matter - total <= -0.08999306:
##                         :...PollutantTotal organic carbon or COD/3 > -0.2645709: [S2]
##                             PollutantTotal organic carbon or COD/3 <= -0.2645709:
##                             :...ReportingThreshold <= -0.03517298:
##                                 :...PollutantCarbon-14(radioactive) > -0.07597014:
##                                 :   :...Easting <= -0.4138854: RS_air (4)
##                                 :   :   Easting > -0.4138854: RS_water (8/1)
##                                  :   PollutantCarbon-14(radioactive) <= -0.07597014: [S3]
##                                  ReportingThreshold > -0.03517298:
##                                  :...ReportingThreshold > -0.03512213:
##                                      :...PollutantToluene > -0.07338001: [S4]
##                                      :   PollutantToluene <= -0.07338001:
##                                      :   :...PollutantChromium > -0.08090754: [S5]
##                                      :       PollutantChromium <= -0.08090754:
##                                      :       :...PollutantNickel > -0.1308462: [S6]
##                                      :           PollutantNickel <= -0.1308462: [S7]
##                                      ReportingThreshold <= -0.03512213:
##                                      :...AmountInT > -0.04863761:
##                                          :...PollutantIron > -0.08556747: Water (19)
##                                          :   PollutantIron <= -0.08556747: [S8]
##                                          AmountInT <= -0.04863761:
##                                          :...PollutantAntimony > -0.04796473: Air (6)
##                                              PollutantAntimony <= -0.04796473: [S9]
##
## SubTree [S2]
##
## RegionTayside, Central and Tayside, Central and Fife > -0.359394: Water (6)
## RegionTayside, Central and Tayside, Central and Fife <= -0.359394:
## :...RegionTayside, Central and Fife > -0.2775826: Water (4)
##     RegionTayside, Central and Fife <= -0.2775826:
##     :...RegionGlasgow and Strathclyde <= -0.5552879:
##         :...Northing <= 0.3209925:
##         :   :...AmountInT <= -0.04829905: Water (3/1)
##         :   :   AmountInT > -0.04829905: W_water (5)
##         :   Northing > 0.3209925:
##         :   :...Easting <= 1.335739: Water (2)
##         :       Easting > 1.335739: W_water (2)
##         RegionGlasgow and Strathclyde > -0.5552879:
##         :...Easting <= -0.5864963: Water (8/1)
##             Easting > -0.5864963:
##             :...Northing > -0.6599158: Water (3)
##                 Northing <= -0.6599158:
##                 :...Easting <= -0.4138854: W_water (3)
##                     Easting > -0.4138854:
##                     :...Northing <= -0.727318: W_water (2)
##                         Northing > -0.727318:
##                         :...Easting <= -0.2391889: Water (3)
##                             Easting > -0.2391889: W_water (3/1)
##
## SubTree [S3]
##
## PollutantIodine-123(radioactive) > -0.05181777: RS_water (5)
## PollutantIodine-123(radioactive) <= -0.05181777:
## :...AmountInT <= -0.04864267: Water (293/10)
##     AmountInT > -0.04864267:
##     :...PollutantDi(2-ethylhexyl) phthalate <= -0.114797: RS_water (29/2)
##         PollutantDi(2-ethylhexyl) phthalate > -0.114797: Water (14)
##
## SubTree [S4]
##
## ReportingThreshold <= -0.03263058: Water (6/1)
## ReportingThreshold > -0.03263058: Air (8)
##
## SubTree [S5]
##
## ReportingThreshold <= -0.0346645: Air (8)
## ReportingThreshold > -0.0346645: Water (9/1)
##
## SubTree [S6]
##
## ReportingThreshold <= -0.0346645: Air (9)
## ReportingThreshold > -0.0346645: Water (35/1)
##
## SubTree [S7]
##
## PollutantChromium-51(radioactive) > -0.05877838: RS_water (5)
## PollutantChromium-51(radioactive) <= -0.05877838:
## :...ReportingThreshold <= -0.03263058:
##     :...PollutantSulphur-35(radioactive) <= -0.06791056: Water (248/11)
##     :   PollutantSulphur-35(radioactive) > -0.06791056: RS_water (4)
##     ReportingThreshold > -0.03263058:
##     :...PollutantZinc <= -0.237886: RS_water (20/3)
##         PollutantZinc > -0.237886: Water (48/1)
##
## SubTree [S8]
##
## PollutantHalogenated organic compounds - total as AOX <= -0.0878073: Air (126)
## PollutantHalogenated organic compounds - total as AOX > -0.0878073: Water (18)
##
## SubTree [S9]
##
## PollutantMercury > -0.105918: Air (5)
## PollutantMercury <= -0.105918:
## :...PollutantArsenic > -0.114797: Air (3)
##     PollutantArsenic <= -0.114797: [S10]
##
## SubTree [S10]
##
## PollutantPolycyclic aromatic hydrocarbons (PAHs) (four indicator compounds of LRTAP) <= -0.07338001:
Water (153/6)
## PollutantPolycyclic aromatic hydrocarbons (PAHs) (four indicator compounds of LRTAP) > -0.07338001:
## :...RegionGlasgow and Strathclyde > -0.5552879: Water (8)
##     RegionGlasgow and Strathclyde <= -0.5552879:
##     :...Northing <= 0.02419023: Air (4)
##         Northing > 0.02419023: W_water (2/1)
##
##
## Evaluation on training data (2613 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##      53   41( 1.6%)   <<
##
##
##     (a)   (b)   (c)   (d)   (e)   (f)    <-classified as
##    ----  ----  ----  ----  ----  ----
##     451                 3           5    (a): class Air
##           546                            (b): class No medium
##                   8     2           1    (c): class RS_air
##                       104           5    (d): class RS_water
##                              15    22    (e): class W_water
##                         1     2  1448    (f): class Water
##
##
```

```
print(swpSkimpy.c5tree$finalModel)
```

```
##
## Call:
## C50:::C5.0.default(x = x, y = y, weights = wts)
##
## Classification Tree
## Number of samples: 2613
## Number of predictors: 88
##
## Tree size: 53
##
## Non-standard options: attempt to group attributes
```

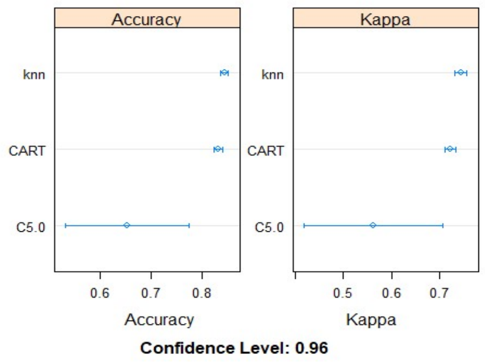
```
confusionMatrix(swpSkimpy.c5tree, norm = "none")
```

```
## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air       1365         0     19      163      57  2366
##   No medium    0      1638      0        0       0     0
##   RS_air       0         0      6        6       0     0
##   RS_water     5         0      7      137       0     1
##   W_water      0         0      0        0       6    16
##   Water        7         0      1       21      48  1970
##
##  Accuracy (average) : 0.6534

#Instance based classifier (k-nn)
set.seed(5)
swpSkimpy.knn<- train(Medium~., data=swpSkimpy,
        method="knn", tuneGrid=expand.grid(k=c(1,3,5,7)),
        #preProcess=c("center","scale"),
        trControl=ctrl)#, metric="Accuracy")

## Aggregating results
## Selecting tuning parameters
## Fitting k = 1 on full training set

plot(swpSkimpy.knn)
```



```
swpSkimpy.knn

## k-Nearest Neighbors
##
## 2613 samples
##    9 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2288, 2285, 2288, 2284, 2287, 2285, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   1  0.8448857  0.7436355
##   3  0.7742071  0.6141452
##   5  0.7460186  0.5562020
##   7  0.7241857  0.5075986
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.

#checking summary results
summary(swpSkimpy.knn$finalModel)
```

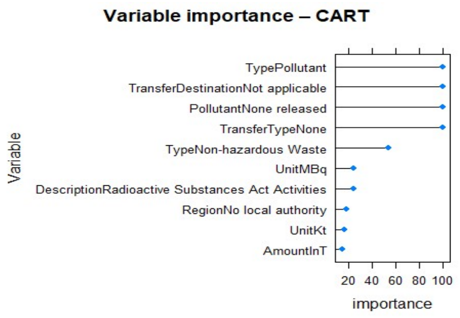
```
##             Length Class      Mode
## learn        2     -none-     list
## k            1     -none-     numeric
## theDots      0     -none-     list
## xNames      88     -none-     character
## problemType  1     -none-     character
## tuneValue    1     data.frame list
## obsLevels    6     -none-     character
## param        0     -none-     list

#confusion matrix
confusionMatrix(swpSkimpy.knn, norm="none")

## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##            Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air       1025       315      2       12      29    85
##   No medium  206      1039     11       28      25    26
##   RS_air       0         2     15        5       0     0
##   RS_water    14        40      4      272       0     3
##   W_water     10        41      0        3      40     7
##   Water      122       201      1        7      17  4232
##
##  Accuracy (average) : 0.8449

# collect resamples (results for comparison on swpSkimpy)
results <- resamples(list(CART=swpSkimpy.cart, C5.0 = swpSkimpy.c5tree, knn= swpSkimpy.knn))
# show accuracy and kappa details
results

##
## Call:
## resamples.default(x = list(CART = swpSkimpy.cart, C5.0 = swpSkimpy.c5tree,
##  knn = swpSkimpy.knn))
##
## Models: CART, C5.0, knn
## Number of resamples: 24
## Performance metrics: Accuracy, Kappa
## Time estimates for: everything, final model fit

summary(results)

##
## Call:
## summary.resamples(object = results)
##
## Models: CART, C5.0, knn
## Number of resamples: 24
##
## Accuracy
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## CART 0.7975460 0.8218654 0.8320607 0.8326267 0.8423438 0.8658537    0
## C5.0 0.3810976 0.3846154 0.6370226 0.6536394 0.9210123 0.9723926    0
## knn  0.8184615 0.8320599 0.8435568 0.8448857 0.8542945 0.8750000    0
##
## Kappa
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## CART 0.6708582 0.7043921 0.7216183 0.7220255 0.7408807 0.7716492    0
## C5.0 0.2445655 0.2471919 0.5353305 0.5620336 0.8752423 0.9540205    0
## knn  0.7020711 0.7236548 0.7408196 0.7436355 0.7600430 0.7949062    0

# Showing results graphically
scales <- list(x=list(relation="free"), y=list(relation= "free"))
dotplot(results, scales=scales, conf.level = 0.96)
```



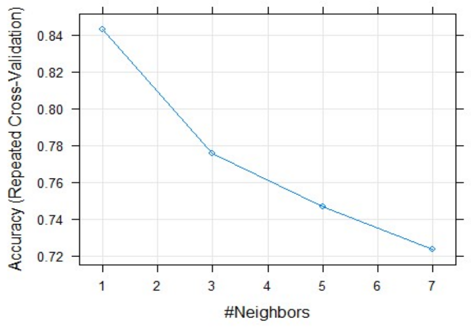
TASK6:

A: Plotted variable importance for mod.cart as it has the highest accuracy compared to C5.0Tree on swpTrain set. The plot shows 100% importance of 4 variables. TransferDestination, Type, Pollutant and TransferType. I have choosed Type, as looking into the dataset the main focus is to analyse and visualise data related to waste and pollutants. I am looking into to dig deeper with Pollutant type. Created swpTrain2 dataset that has total 5 copies(duplicates) of column "Type" B: Running K-NN model (instance based) on swpTrain2 dataset with tuning,ctrl and metric as task 3. We have accuracy 0.84 and kappa 0.74 for k=1. Looking into the confusion matrix, there are lots of wrongly classified classes when compared to model knn in task 3. With comparative analysis on scale of confidence level 0.96 swpTrain2 with knn has very very small amount of increase in accuracy

```
# A:
# plot 10 most important variables in the model mod.cart as it has highest accuracy compared to mod.c5tree
plot(varImp(mod.cart), top=10, main="Variable importance - CART", xlab="importance", ylab= "Variable")
```

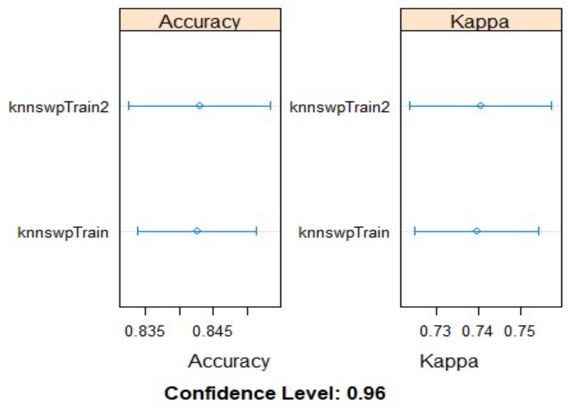


```
# Replicating "Type" 4 times in swpTrain2 using swpTrain
swpTrain2 <- swpTrain
swpTrain2$Duplicated1 <- swpTrain2$Type
swpTrain2$Duplicated2 <- swpTrain2$Type
swpTrain2$Duplicated3 <- swpTrain2$Type
swpTrain2$Duplicated4 <- swpTrain2$Type

# B: Instance based classifier on swpTrain2 used in task3 K-nn
set.seed(6)
swpTrain2.knn<- train(Medium~., data=swpTrain2,
        method="knn", tuneGrid=expand.grid(k=c(1,3,5,7)),
        #preProcess=c("center","scale"),
        trControl=ctrl, metric="Accuracy")

## Aggregating results
## Selecting tuning parameters
## Fitting k = 1 on full training set

plot(swpTrain2.knn)
```



```
swpTrain2.knn

## k-Nearest Neighbors
##
## 2613 samples
##   15 predictor
##    6 classes: 'Air', 'No medium', 'RS_air', 'RS_water', 'W_water', 'Water'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2287, 2285, 2287, 2286, 2285, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   1  0.8429743  0.7405025
##   3  0.7757559  0.6179919
##   5  0.7466789  0.5573656
##   7  0.7234568  0.5078771
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 1.

#checking summary results
summary(swpTrain2.knn$finalModel)

##             Length Class      Mode
## learn         2    -none-     list
## k             1    -none-     numeric
## theDots       0    -none-     list
## xNames      109    -none-     character
## problemType   1    -none-     character
## tuneValue     1    data.frame list
## obsLevels     6    -none-     character
## param         0    -none-     list

#confusion matrix
confusionMatrix(swpTrain2.knn, norm="none")

## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##             Reference
## Prediction   Air No medium RS_air RS_water W_water Water
##   Air       1018       329      5       11      28    91
##   No medium  216      1034      7       26      27    27
##   RS_air       0         1     14        6       0     0
##   RS_water    13        35      5      271       0     4
##   W_water     11        34      0        4      42     2
##   Water      119       205      2        9      14  4229
##
##  Accuracy (average) : 0.843
```

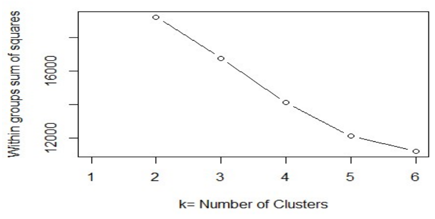
```
# collect resamples (results for comparison on swpTrain2)
results <- resamples(list(knnswpTrain=mod.knn, knnswpTrain2 = swpTrain2.knn))
# show accuracy and kappa details
results

##
## Call:
## resamples.default(x = list(knnswpTrain = mod.knn, knnswpTrain2 = swpTrain2.knn))
##
## Models: knnswpTrain, knnswpTrain2
## Number of resamples: 24
## Performance metrics: Accuracy, Kappa
## Time estimates for: everything, final model fit

summary(results)

##
## Call:
## summary.resamples(object = results)
##
## Models: knnswpTrain, knnswpTrain2
## Number of resamples: 24
##
## Accuracy
##                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## knnswpTrain  0.8103976 0.8311773 0.8442744 0.8426156 0.8535276 0.8926380    0
## knnswpTrain2 0.7975460 0.8228995 0.8435583 0.8429743 0.8540901 0.8899083    0
##
## Kappa
##                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## knnswpTrain  0.6857329 0.7168905 0.7443290 0.7395923 0.7594439 0.8239578    0
## knnswpTrain2 0.6686278 0.7075565 0.7385397 0.7405025 0.7588140 0.8159590    0

# Showing results graphically
scales <- list(x=list(relation="free"), y=list(relation= "free"))
dotplot(results, scales=scales, conf.level = 0.96)
```



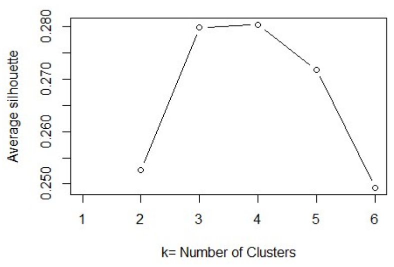
TASK7:

Cannot read row names due to increase in dimensionality. Firstly tried deleting categorical variables but clusters was not giving any correlation with attributes when performing CART for task8. As without categorical variables it is difficult to get correlation with attributes although it is possible to find correlated instances in each cluster. Finding optimal number of clusters between 2 to 6 plot indicates k=3 at an elbow. Silhouette is calculated for each k means that gives the k=4. Applying k means with Lloyd, Forgy and MacQueen gives the same results. Majority of clusters is missed together. too with the above. Cluster 1 has 1588 instances while Cluster 2 with 544 and Cluster 3 with 481 instances. Clusters are far from each other and at very distance hardly visible in graphs. But we can plot their size as below. Cluster 1 has 235 instances Cluster 2 has 1409 instances Cluster 3 has 530 instances Cluster 4 has 439 instances. Methods Llyod, Forgy and MacQueen all are giving the same results.

```
dim(swpTrain)

## [1] 2613   12

pca_swpTrain <- preProcess(swpTrain, method = c("pca")) #Applying pca
DF_swpTrain <- predict(pca_swpTrain, newdata = swpTrain)  # Creating dataset with pc variables
binaryVars <- dummyVars(~ ., data = DF_swpTrain) #one hot encoding character variables
DF_swpTrain2 <- predict(binaryVars, newdata = DF_swpTrain) #Matrix of correlation for cluster anlysis

#Finding k means
set.seed(7)
wss <- NULL
for (i in 2:6)
  wss[i] <- sum(kmeans(DF_swpTrain2, centers=i, nstart=100, iter.max=1000)$withinss)
plot(1:6, wss, type="b", xlab="k= Number of Clusters", ylab="Within groups sum of squares")
```



```
#Calculating average silhouette
set.seed(7)
sil <-NULL
for (i in 2:6)
  {
  res <- kmeans(DF_swpTrain2, centers = i, nstart = 25)
  ss <- silhouette(res$cluster, dist(DF_swpTrain2))
  sil[i] <- mean(ss[, 3])
  }
plot(1:6, sil, type="b", xlab="k= Number of Clusters", ylab="Average silhouette")
```



```
#Applying kmeans with K=4 Lloyd
set.seed(7)
km <- kmeans(DF_swpTrain2, 4, algorithm = "Lloyd", nstart=25, iter.max=1000)

#Viewing results
#palette(alpha(brewer.pal(9,'Set1'), 0.5))
#plot(DF_swpTrain2, col=km$cluster, pch=16)
#plot3d(DF_swpTrain$PC1, DF_swpTrain$PC2, DF_swpTrain$PC3, col=km$cluster)
#plot3d(DF_swpTrain$PC1, DF_swpTrain$PC2, DF_swpTrain$PC4, col=km$cluster)

#Looking at the size of clusters
sort(table(km$cluster))
```

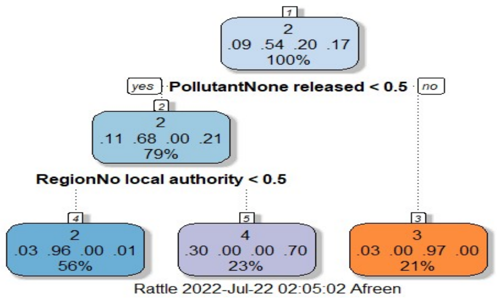
```
##
##    1    4    3    2
##  235  439  530 1409

clust <- names(sort(table(km$cluster)))

#Getting instances in each cluster
#row.names(swpTrain2[km$cluster==clust[1],1:3])
#Instances of cluster2
#row.names(swpTrain2[km$cluster==clust[2],4:6])
#Instances of cluster3
#row.names(swpTrain2[km$cluster==clust[3],7:9])
#Instances of cluster4
#row.names(swpTrain2[km$cluster==clust[4],10:12])

#Correlation between  columns of each cluster
#cor(DF_swpTrain2)
```

TASK8:

A.Dataset generated in task7 DF_SWPTrain has been used for this task that has total 12 variables. New column created as Cluster which represents a cluster name of each instance to which it belongs as factor. swpClust has total 2465 observations and 13 attributes. B. Applying CART rpart with same ctrl used in task3 repeatedcv 8 cross validation and repeats 3. Accuracy 0.94, Kappa 0.90. Prediction says cluster 4 has been most correctly classified with correct related clusters according to confusion matrix. Region No local authority, RegionNo local authority, TypePollutant, TransferDestinationNot applicable, MediumNo medium, PollutantNone released and TransferTypeNone are the most important variables

```
# A:
#Obtaining column with cluster number of specific instance
swpClust <- cbind(DF_swpTrain, Cluster = as.factor(km$cluster))

#B:
#CART
pca.cart<- train(Cluster~., data = swpClust, method="rpart", #preProcess=c("center","scale"),
                 metric="Accuracy", trControl=ctrl)


## Aggregating results
## Selecting tuning parameters
## Fitting cp = 0.149 on full training set

pca.cart

## CART
##
## 2613 samples
##   12 predictor
##    4 classes: '1', '2', '3', '4'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold, repeated 3 times)
## Summary of sample sizes: 2286, 2286, 2286, 2286, 2287, 2287, ...
## Resampling results across tuning parameters:
##
##   cp         Accuracy   Kappa
##   0.1486711  0.9438631  0.9092277
##   0.3504983  0.8072504  0.6478080
##   0.4393688  0.6314301  0.2340890
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.1486711.

varImp(pca.cart)
```

```
## rpart variable importance
##
##   only 20 most important variables shown (out of 111)
##
##                                                           Overall
## RegionNo local authority                                   100.00
## TypePollutant                                               99.87
## TransferDestinationNot applicable                           99.87
## MediumNo medium                                             99.87
## PollutantNone released                                      99.87
## TransferTypeNone                                            99.63
## PC4                                                         93.09
## PC3                                                         79.96

#summary(mod.cart)
fancyRpartPlot(pca.cart$finalModel)
```



```
confusionMatrix(pca.cart, norm = "none")

## Cross-Validated (8 fold, repeated 3 times) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##           Reference
## Prediction    1    2    3    4
##          1  319    0    0    0
##          2  117 4227    3   51
##          3   51    0 1587    0
##          4  218    0    0 1266
##
##  Accuracy (average) : 0.9439
```

REFERENCE

Anon., n.d. [Online] Available at: https://towardsdatascience.com/clustering-datasets-having-both-numerical-and-categorical-variables-ed91cdca0677 Anon., n.d. [Online] Available at: https://www.statmethods.net/advstats/cluster.html Anon., n.d. [Online] Available at: W3Schools.com Anon., n.d. Quick R by data camp. [Online] Available at: https://www.statmethods.net/advstats/cluster.html Lecture notes - Innes Arana, n.d. Clustering. s.l.:s.n. lecture notes - Innes Arana, n.d. naive bayes demo. s.l.:s.n. Brownlee, J., n.d. Step by step machine learning. [Online] Available at: https://machinelearningmastery.com/machine-learning-in-r-step-by-step/