COURSEWORK CMM510 (Data Mining)

School of Computing Science and Digital Media

Robert Gordon University

Juweria Wajid Ali

Student ID: 2015099

Submission Date: 30/11/2021

Contents

Data from R-Studio

##Clearing the workspace and setting the working directory.

```
rm(list=ls())
```

#Setting the working directory

```
library(here)

setwd(here::here())
```

#loading libraries

```
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(Hmisc)

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(C50)
library(mlbench)
library(RColorBrewer)
library(scales)
library(cluster)
library(rgl)
library(fpc)
library(pvclust)
```

#Loading & reading the dataset ride4U

```
ride4U <- read.csv("ride4U.csv", header=T, stringsAsFactors=T)
```

#Get high level view of data

```
str(ride4U)
```

```
## 'data.frame':    730 obs. of  13 variables:
##  $ city         : Factor w/ 3 levels "Gordontown","robertburgh",..: 1 3 1
3 1 3 1 1 1 1 ...
##  $ country      : Factor w/ 1 level "Universitalia": 1 1 1 1 1 1 1 1 1 1 .
..
##  $ month        : Factor w/ 12 levels "April","August",..: 8 5 6 4 5 8 5 4
10 3 ...
##  $ day          : int  9 7 13 9 20 27 4 10 7 12 ...
##  $ holiday      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ day_of_week  : Factor w/ 7 levels "Friday","Monday",..: 3 4 3 1 4 6 1 4
5 5 ...
##  $ outlook      : Factor w/ 4 levels "","cloudy","rainy",..: 2 2 2 2 2 2 2
2 2 2 ...
##  $ temperature  : num  16.65 7.4 29.99 6.43 9.71 ...
##  $ humidity     : num  38.8 45.2 45.6 46.9 47.5 ...
##  $ feel_humidity: Factor w/ 3 levels "comfortable",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ wind         : Factor w/ 5 levels "calm","gale",..: 5 3 1 3 4 3 3 3 5 3
...
##  $ uses         : int  5438 1803 8849 1924 3786 1990 2829 4577 6004 6341 .
..
##  $ complaints   : Factor w/ 3 levels "lots","some",..: 2 3 2 3 3 3 3 3 2 2
...
```

```
summary(ride4U)
```

```
##           city              country          month          day          holi
day
##  Gordontown :365   Universitalia:730   August   : 62   Min.   : 1.00   No :
706
```

```
##  robertburgh:  2                       December: 62   1st Qu.: 8.00   Yes:
24
##  Robertburgh:363                        January : 62   Median :16.00
##                                         July    : 62   Mean   :15.72
##                                         March   : 62   3rd Qu.:23.00
##                                         May     : 62   Max.   :31.00
##                                         (Other) :358
##     day_of_week    outlook     temperature       humidity         feel_hum
idity
##  Friday   :104         :  3   Min.   : 1.82   Min.   :24.21   comfortable:
327
##  Monday   :105   cloudy:246   1st Qu.:13.79   1st Qu.:54.13   dry        :
25
##  Saturday :104   rainy : 21   Median :20.46   Median :64.90   humid      :
378
##  Sunday   :104   sunny :460   Mean   :20.33   Mean   :65.16
##  Thursday :104                3rd Qu.:26.84   3rd Qu.:75.88
##  Tuesday  :105                Max.   :35.80   Max.   :97.60
##  Wednesday:104                NA's   :1       NA's   :1
##        wind          uses          complaints
##  calm    : 85   Min.   :   35   lots    : 11
##  gale    :  1   1st Qu.: 3718   some    :403
##  light   :336   Median : 5370   very_few:316
##  moderate:197   Mean   : 5322
##  strong  :111   3rd Qu.: 7008
##                 Max.   :10150
##                 NA's   :3
```

```
class(ride4U)
```

```
## [1] "data.frame"
```

```
View(ride4U)
head(ride4U)
```

```
##          city       country    month day holiday day_of_week outlook
## 1  Gordontown Universitalia    March   9      No    Saturday  cloudy
## 2 Robertburgh Universitalia  January   7      No      Sunday  cloudy
## 3  Gordontown Universitalia     July  13      No    Saturday  cloudy
## 4 Robertburgh Universitalia February   9      No      Friday  cloudy
## 5  Gordontown Universitalia  January  20      No      Sunday  cloudy
## 6 Robertburgh Universitalia    March  27      No     Tuesday  cloudy
##   temperature humidity feel_humidity     wind uses complaints
## 1       16.65    38.76   comfortable   strong 5438       some
## 2        7.40    45.23   comfortable    light 1803   very_few
## 3       29.99    45.63   comfortable     calm 8849       some
## 4        6.43    46.90   comfortable    light 1924   very_few
## 5        9.71    47.52   comfortable moderate 3786   very_few
## 6        9.92    48.32   comfortable    light 1990   very_few
```

Task 1: To Inspect ride4U Dataset

The ride4U dataset contains 730, instances, 4 numeric attributes (day, temperature, humidity, uses) and 8 nominal attributes (city, country, month, holiday, day_of_week, outlook, feel_humidity, wind) and the nominal class attribute complaints with three values (lots, some and very few).

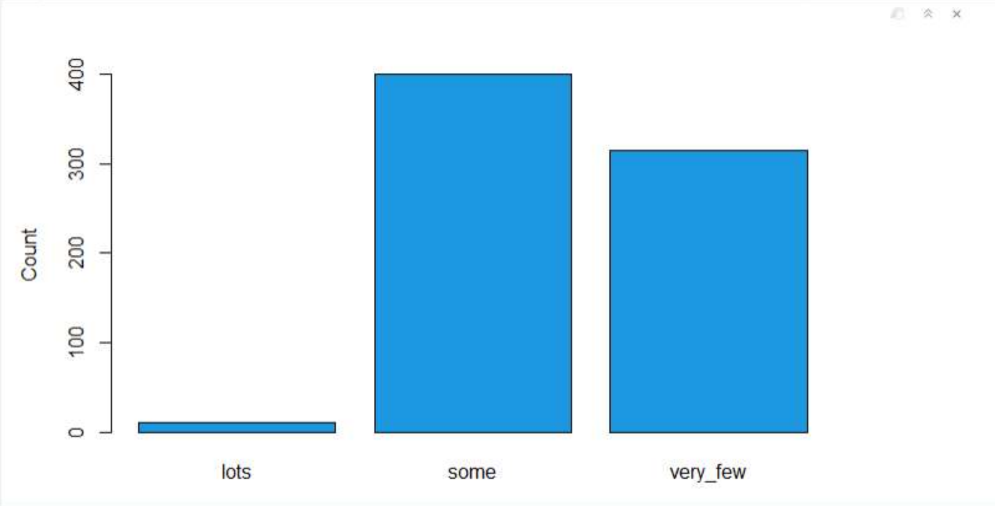
Analysis of data refers to studying the characteristics of the object under study, and for determining the patterns of relationships among the variables relating to it.

1a) Univariate Analysis

It is a method for analyzing the data on a single variable at a time, where we are observing only one aspect of the phenomenon at a time.

Class distribution:

````
##Checking class distribution.
```{r}
# checking class distribution
plot(ride4U$complaints, xlab ="Class label", ylab ="Count",col ="#1b98e0")
```
````



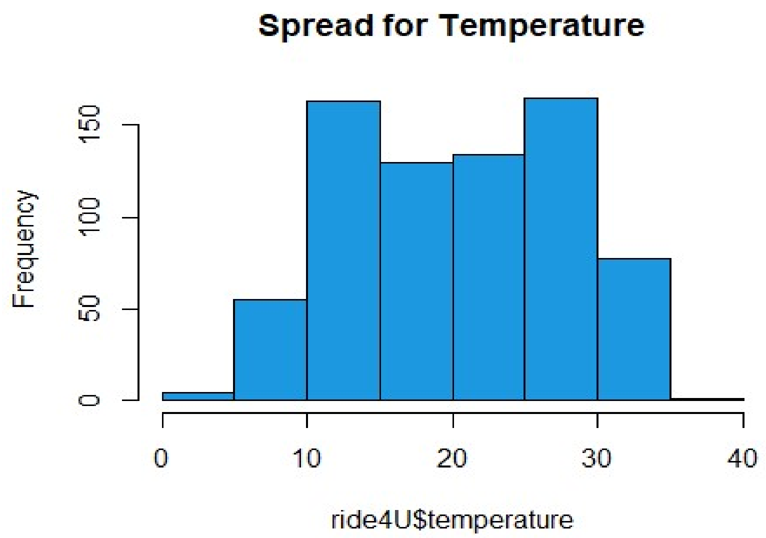
The plot shows about 400 instances of class 'some', 315 instances of class 'very few' and about 10 instances of class 'lots'.

Temperature: continuous variable

Histogram for temperature

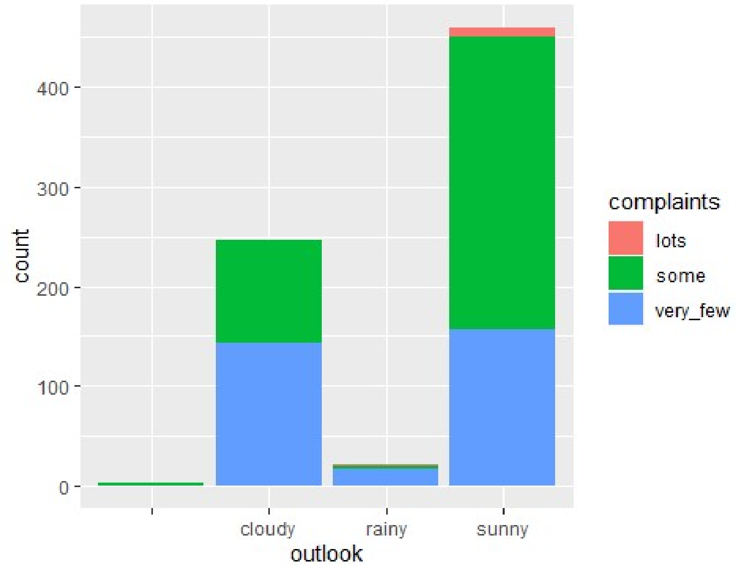
```
summary(ride4U$temperature)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.82   13.79   20.46   20.33   26.84   35.80       1

hist(ride4U$temperature, main = "Spread for Temperature",col = "#1b98e0")
```

Here we have temperature on the x-axis and frequency on the y-axis. Looking at the analysis and graph we can say that the minimum temperature is 1.82 and maximum is 35.80. We can also see the mean i.e. the average temperature is 20.33.



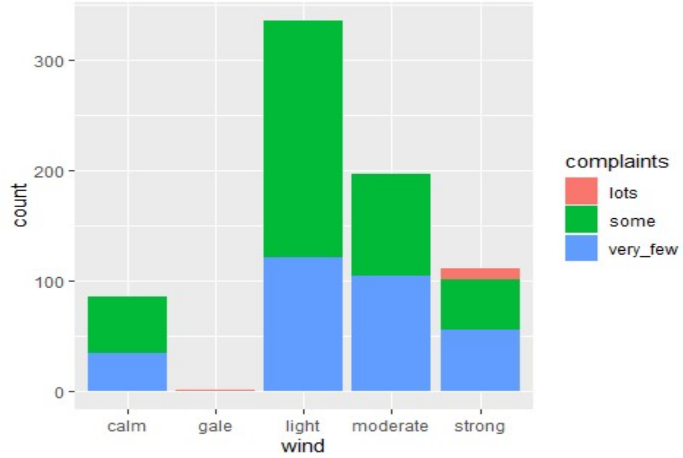
1b) Bivariate Analysis
Here we look at finding relationships between variables, we are not concerned about "why", and we just look at "what". The type of graph usually used here is stacked bar plot.

```
ggplot(ride4U,
       aes(x = outlook,
           fill = complaints)) +
  geom_bar(position = "stack")
```



We see here that outlook is mostly sunny, followed by cloudy in the second place. And, there is only a small difference in the level of complaints when the outlook is cloud and sunny.

```
ggplot(ride4U,
       aes(x = wind,
           fill = complaints)) +
  geom_bar(position = "stack")
```



Similarly, here we see the wind is mostly light. Complaints are a lot when there is gale wind, and when the wind is strong.

Task 2: Preparing the Data

- When files are loaded, nominal (categorical) attributes need to be factors, so use stringsAsFactors=T when you load the file (Arana 2021).
- Name unification: It was observed in the dataset that attribute city has two values for Robertburgh and robertburgh which is one and the same, only the fact that the alphabet cases differ. We can unify these names to one i.e. “Robertburgh” to make it uniform. The instance 54, 728 is “robertburgh” which we are going to change to “Robertburgh”.

```
ride4U$city <- as.character(ride4U$city)

ride4U$city <- if_else(ride4U$city == 'robertburgh','Robertburgh', ride
4U$city)
```

- Addressing and dealing with missing values: As in our dataset we do not have many instances with missing values they can be disregarded or, they can be made to pass as it is and the algorithms deal with them later on

```
ride4U$city <- as.factor(ride4U$city)
ride4U<- na.omit(ride4U)
ride4U<-subset(ride4U,select=-c(country))
```

Task 3: Obtaining further reduced datasets

3a) ride4U40: 40% of ride4U dataset

```
set.seed(123)
indexride4U40 <- createDataPartition(y = ride4U$complaints,
                                     p = .4,
                                     list = FALSE)
ride4u40 <- ride4U[indexride4U40,]
```

3b) ride4U20: 50% of ride4U40 dataset

```
set.seed(123)
indexride4U20 <- createDataPartition(y = ride4u40$complaints,
                                     p = .5,
                                     list = FALSE)
ride4u20 <- ride4u40[indexride4U20,]
```

3c) dodgyRide4U with 15% noise in attributes outlook and temperature

##making a copy of the dataset

```
set.seed(123)
dodgyRide4U<- ride4U
corrupt1 <- rbinom(nrow(ride4U),1,0.15)
corrupt1 <- as.logical(corrupt1)
```

Introducing noise in selected instances for outlook attribute

```
set.seed(123)
noise1 <- sample(dodgyRide4U$outlook, length(dodgyRide4U$outlook) - 1, replac
e = T)
dodgyRide4U$outlook[corrupt1] <- noise1[corrupt1]
```

Introducing noise in numeric attribute - temperature

```
set.seed(123)
noise2 <- rnorm(corrupt1, median(dodgyRide4U$temperature), sd(dodgyRide4U$tem
perature))
dodgyRide4U$temperature[corrupt1] <- as.integer(noise2[corrupt1])
```

Discussion-Reduced Datasets
Advantages:

- Reducing data may give us a much compressed description of the data i.e. in quantity but without compromising on the quality of original data.

- Reduced size of datasets take up less memory space. Hence, improves storage efficiency and reduces storage costs.
- Less data, better computation/training time.
- Improved visualisation of data.
- Data reduction does not affect the result obtained from data mining, i.e. the result obtained from data mining before and after data reduction is almost the same. The only difference occurs in the efficiency of data mining which increases (T 2020).

Disadvantage:

- A disadvantage may be some loss of information in the reduced data sets.

Noisy Data

In real world, there are many factors that affect the data, noise is one of them. The presence of noise has an impact on the prediction of information that is meaningful. A lot of studies have shown that noise will have an effect on the classification accuracy which may be decreased that results in poor results of prediction (Gupta 2019).

Simple predictions or assumptions to explain the data may not be sufficient and we may have to come up with more complex assumptions. This also impacts on the use of computing resources, which is increase (Educate 2020).

Task 4: Experiment with different dataset sizes

First create the models, then compare the results.

C5.0 model creation

4a) C5.0Tree model on full size dataset i.e. ride4U

```
set.seed(123)
treeAll <- train(complaints ~ .,
    data = ride4U,
    method = "C5.0Tree",
    metric = "Accuracy",
    trControl = trainControl(method = "cv"))
```

##To view results

```
summary(treeAll$finalModel)

##
## Call:
## C50:::C5.0.default(x = x, y = y, weights = wts)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Nov 29 23:56:50 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 726 cases (33 attributes) from undefined.data
##
## Decision tree:
##
## uses > 4998:
## :...holidayYes <= 0: some (400)
## :   holidayYes > 0: very_few (8)
## uses <= 4998:
## :...temperature > 8.89: very_few (272/1)
##     temperature <= 8.89:
##     :...windstrong <= 0: very_few (36)
##         windstrong > 0: lots (10)
##
##
## Evaluation on training data (726 cases):
##
##       Decision Tree
##     ----------------
##     Size      Errors
##
##        5    1( 0.1%)   <<
##
```

```
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##      10           1    (a): class lots
##          400           (b): class some
##                 315    (c): class very_few
##
##
##  Attribute usage:
##
##  100.00% uses
##   56.20% holidayYes
##   43.80% temperature
##    6.34% windstrong
##
##
## Time: 0.0 secs
```

##Confusion matrix

```
confusionMatrix(treeAll)

## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction lots some very_few
##   lots      1.2  0.0      0.0
##   some      0.0 55.1      0.4
##   very_few  0.3  0.0     43.0
##
##  Accuracy (average) : 0.9931
```

4b) C5.0Tree model on rid4U40

```
set.seed(123)
tree40 <- train(complaints ~ .,
    data = ride4u40,
    method = "C5.0Tree",
    metric = "Accuracy",
    trControl=trainControl(method="cv"))
```

##To view results

```
summary(tree40$finalModel)

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

```
##
## C5.0 [Release 2.07 GPL Edition]      Mon Nov 29 23:56:55 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 291 cases (33 attributes) from undefined.data
##
## Decision tree:
##
## uses <= 4998: very_few (128/5)
## uses > 4998:
## :...holidayYes <= 0: some (160)
##     holidayYes > 0: very_few (3)
##
##
## Evaluation on training data (291 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       3    5( 1.7%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##                   5    (a): class lots
##           160          (b): class some
##                 126    (c): class very_few
##
##
##  Attribute usage:
##
##  100.00% uses
##   56.01% holidayYes
##
##
## Time: 0.0 secs
```

##Confusion matrix

```
confusionMatrix(tree40)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction lots some very_few
```

```
##   lots       0.3  0.0       0.0
##   some       0.0 55.0       1.0
##   very_few   1.4  0.0      42.3
##
##  Accuracy (average) : 0.9759
```

4c) C5.0Tree model on ride4U20

```
set.seed(123)
tree20 <- train(complaints ~ .,
    data = ride4u20,
    method = "C5.0Tree",
    metric = "Accuracy",
    trControl=trainControl(method="cv"))

##To view results
summary(tree20$finalModel)

##
## Call:
## C50:::C5.0.default(x = x, y = y, weights = wts)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Nov 29 23:57:00 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 146 cases (33 attributes) from undefined.data
##
## Decision tree:
##
## uses > 4898:
## :...holidayYes <= 0: some (80)
## :   holidayYes > 0: very_few (3)
## uses <= 4898:
## :...day_of_weekThursday <= 0: very_few (58/1)
##     day_of_weekThursday > 0:
##     :...windstrong <= 0: very_few (3)
##         windstrong > 0: lots (2)
##
##
## Evaluation on training data (146 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       5    1( 0.7%)   <<
```

```
##
##
##      (a)   (b)   (c)    <-classified as
##     ----  ----  ----
##        2           1    (a): class lots
##             80          (b): class some
##                   63    (c): class very_few
##
##
##  Attribute usage:
##
##  100.00% uses
##   56.85% holidayYes
##   43.15% day_of_weekThursday
##    3.42% windstrong
##
##
## Time: 0.0 secs
```

##Confusion matrix

```
confusionMatrix(tree20)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##            Reference
## Prediction lots some very_few
##   lots      0.0  0.0      0.0
##   some      0.0 54.8      1.4
##   very_few  2.1  0.0     41.8
##
##  Accuracy (average) : 0.9658
```

Evaluation and discussion

Measure	Accuracy(avg)
Dataset	
ride4U	0.9931
ride4U40	0.9759
ride4U20	0.9658

C5.0Tree on ride4u dataset correctly classifies all instances of the three classes, except one of the 'very few' instance which is misclassified. Class 'lots' and class 'some' are correctly identified in all cases, and there are no false positives.

C5.0Tree on ride4U40 correctly classifies all instances of class 'some' and all instances of class 'very few', except five which were misclassified. There was no instance of class 'lots' correctly classified.

C5.0Tree on ride4U20 correctly classifies all instances of the three classes except one that is misclassified (Arana 2021).

The similarity between all three models is that the most preferred class is 'some' because it has no instances misclassified.

The accuracy for the full dataset ride4U is 0.9931.When the size of the dataset is reduced to 40% we can see a decrease in accuracy, however when this reduced dataset is further reduced, there is not a very significant change in the accuracy. Looking at the table above and using accuracy as the measure we may say that the C5.0Tree model on ride4U dataset performs the best. We may also say that in our case, reduction in size of the datasets does lead to a reduction in performance in some way.

Task 5: Experiment with Noisy Dataset

5a) Tree classifier - for ride4U and dodgyRide4U

ride4U dataset

```
set.seed(123)
treeAll <- train(complaints ~ .,
    data = ride4U,
    method = "C5.0Tree",
    metric = "Accuracy",
    trControl = trainControl(method="cv"))
```

##To view results

```
summary(treeAll$finalModel)

##
## Call:
## C50:::C5.0.default(x = x, y = y, weights = wts)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Nov 29 23:57:06 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 726 cases (33 attributes) from undefined.data
##
## Decision tree:
##
## uses > 4998:
## :...holidayYes <= 0: some (400)
## :   holidayYes > 0: very_few (8)
## uses <= 4998:
## :...temperature > 8.89: very_few (272/1)
##     temperature <= 8.89:
##     :...windstrong <= 0: very_few (36)
##         windstrong > 0: lots (10)
##
##
## Evaluation on training data (726 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       5    1( 0.1%)   <<
##
##
```

```
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##      10           1    (a): class lots
##           400          (b): class some
##                 315    (c): class very_few
##
##
##  Attribute usage:
##
##  100.00% uses
##   56.20% holidayYes
##   43.80% temperature
##    6.34% windstrong
##
##
## Time: 0.0 secs
```

##Confusion matrix

```
confusionMatrix(treeAll)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction lots some very_few
##   lots      1.2  0.0      0.0
##   some      0.0 55.1      0.4
##   very_few  0.3  0.0     43.0
##
##  Accuracy (average) : 0.9931
```

##dodgyRide4U dataset

```
set.seed(123)
treedodgyRide4U <- train(complaints ~ .,
    data = dodgyRide4U,
    method = "C5.0Tree",
    metric = "Accuracy",
    trControl=trainControl(method="cv"))
```

##To view results

```
summary(treedodgyRide4U$finalModel)
```

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

```
##
## C5.0 [Release 2.07 GPL Edition]      Mon Nov 29 23:57:13 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 726 cases (33 attributes) from undefined.data
##
## Decision tree:
##
## uses > 4998:
## :...holidayYes <= 0: some (400)
## :   holidayYes > 0: very_few (8)
## uses <= 4998:
## :...windstrong <= 0: very_few (253/1)
##     windstrong > 0:
##     :...temperature <= 8.42: lots (7/1)
##         temperature > 8.42: very_few (58/4)
##
##
## Evaluation on training data (726 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       5    6( 0.8%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##       6           5    (a): class lots
##           400          (b): class some
##       1         314    (c): class very_few
##
##
##  Attribute usage:
##
##  100.00% uses
##   56.20% holidayYes
##   43.80% windstrong
##    8.95% temperature
##
##
## Time: 0.0 secs
```

##Confusion matrix

```
confusionMatrix(treedodgyRide4U)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction lots some very_few
##   lots      0.8  0.0      0.4
##   some      0.0 55.1      0.4
##   very_few  0.7  0.0     42.6
##
##  Accuracy (average) : 0.9848
```

5b) Instance based classifier – k-nn for ride4U and dodgyRide4U

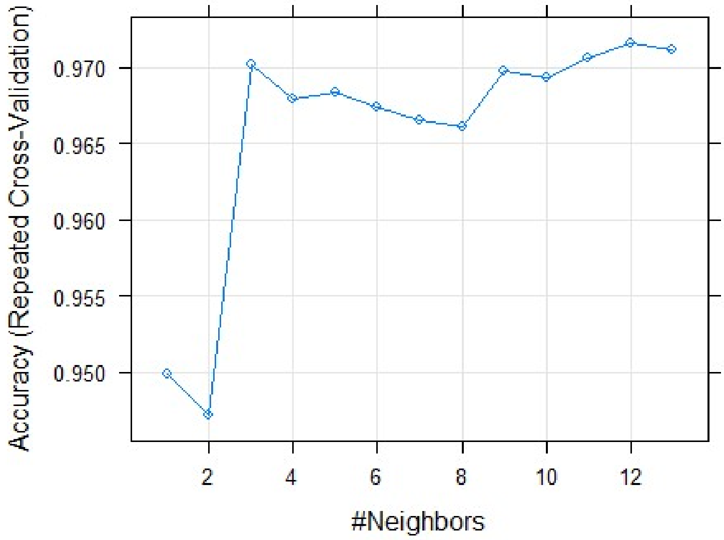
Using 10-fold cross validation for ride4U

```
ctrl1 <- trainControl(method="repeatedcv", number=10, repeats=3)
```

With k values of up to k=13

```
set.seed(123)
mod1 <- train(complaints~., data=ride4U, method="knn", tuneGrid=expand.grid(.
k=1:13), trControl=ctrl1)
print(mod1)

## k-Nearest Neighbors
##
## 726 samples
##  11 predictor
##   3 classes: 'lots', 'some', 'very_few'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 653, 654, 653, 653, 653, 654, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9499429  0.9019278
##    2  0.9472159  0.8963524
##    3  0.9701674  0.9409760
##    4  0.9678716  0.9363529
##    5  0.9683346  0.9372405
##    6  0.9674214  0.9352683
##    7  0.9665081  0.9335029
##    8  0.9660388  0.9325274
##    9  0.9697235  0.9396497
##   10  0.9692669  0.9386901
##   11  0.9706367  0.9413869
##   12  0.9715563  0.9432113
```

```
##   13  0.9710997  0.9423064
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 12.

plot(mod1)
```




```
confusionMatrix.train(mod1, norm="average")

## Cross-Validated (10 fold, repeated 3 times) Confusion Matrix
##
## (entries are average cell counts across resamples)
##
##            Reference
## Prediction lots some very_few
##   lots      0.0  0.0      0.0
##   some      0.0 39.8      0.8
##   very_few  1.1  0.2     30.7
##
##  Accuracy (average) : 0.9715
```

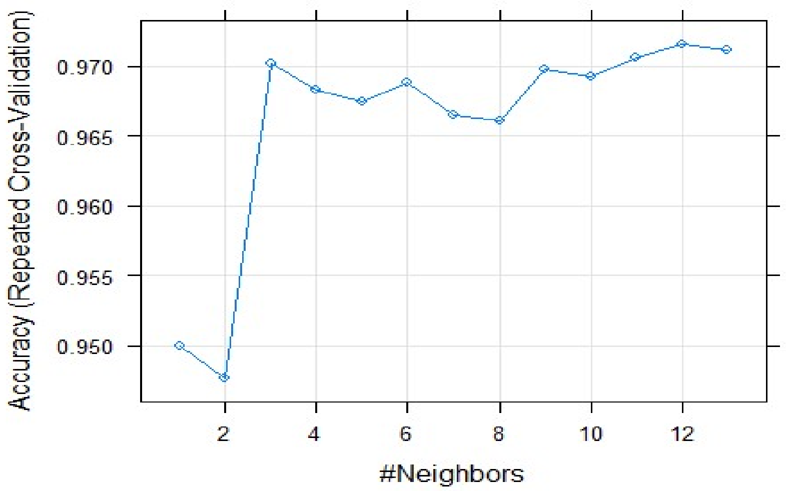
Using 10-fold cross validation for dodgyRide4U

##As above but with k values of up to k=13 for dodgyRide4U

```
set.seed(123)
mod2 <- train(complaints~., data=dodgyRide4U, method="knn", tuneGrid=expand.g
rid(.k=1:13), trControl=ctrl1)
print(mod2)
```

```
## k-Nearest Neighbors
##
## 726 samples
##  11 predictor
##   3 classes: 'lots', 'some', 'very_few'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 653, 654, 653, 653, 653, 654, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9499429  0.9019892
##    2  0.9476788  0.8972235
##    3  0.9701674  0.9409760
##    4  0.9683156  0.9371832
##    5  0.9674214  0.9354198
##    6  0.9688039  0.9379593
##    7  0.9665081  0.9335029
##    8  0.9660388  0.9324965
##    9  0.9697235  0.9396497
##   10  0.9692669  0.9386901
##   11  0.9706367  0.9413869
##   12  0.9715563  0.9432113
##   13  0.9710997  0.9423064
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 12.
```

```
plot(mod2)
```




```
confusionMatrix.train(mod2, norm="average")
```

```
## Cross-Validated (10 fold, repeated 3 times) Confusion Matrix
##
## (entries are average cell counts across resamples)
##
##            Reference
## Prediction lots some very_few
##   lots      0.0  0.0      0.0
##   some      0.0 39.8      0.8
##   very_few  1.1  0.2     30.7
##
##  Accuracy (average) : 0.9715
```

Evaluation and discussion

Measure/Method	Accuracy(avg)/C5.0Tree	Error/C5.0Tree	Accuracy(avg)/K-nearest neighbor
Dataset			
ride4U	0.9931	1(0.1%)	0.9715/best k 12
dodgyRide4U	0.9848	6(0.8%)	0.9715/best k 12

C5.0Tree on ride4u dataset correctly classifies all instances of the three classes, except one of the 'very few' instance which is misclassified. Class 'lots' and class 'some' are correctly identified in all cases, and there are no false positives.

C5.0Tree on dodgyRide4u dataset correctly classifies all instances of the three classes, except six instances which are misclassified. Class 'some' is correctly identified in all cases, and there are no false positives. Class 'lots' has one instance misclassified whereas class 'very few' has five instances misclassified (Arana 2021).

Looking at the table above, we may say that although the accuracy has not changed significantly, there is a significant increase in the errors in the dataset with noise. Hence, we may conclude that introducing noise leads to a reduction in performance of the tree classifier used in this task.

Performing an instance based classifier experiment on both datasets gives us the same result in terms of accuracy and the best k-nn value which is 12.

We may conclude from this experiment that the instance based classifier i.e. K-nearest neighbor works best on datasets with noise when compared to the tree classifier where performance was reduced on the noisy dataset.

Task 6: Testing on ride4UT Dataset

loading & reading the dataset ride4U

```
ride4UT <- read.csv("ride4UT.csv", header=T, stringsAsFactors=T)
View (ride4UT)
```

##dealing with missing values and eliminating useless attribute country

```
ride4UT<- na.omit(ride4UT)
ride4UT<-subset(ride4UT,select=-c(country))
ride4UT$city <- as.character(ride4UT$city)
ride4UT$city <- if_else(ride4UT$city == 'robertburgh','Robertburgh', ride4UT$
city)
ride4UT$city <- as.factor(ride4UT$city)
```

6a) Testing C5.0Tree on ride4UT

```
TestRestreeAll <- predict(treeAll, newdata = ride4UT, type="raw")

confusionMatrix(TestRestreeAll, ride4UT$complaints)

## Confusion Matrix and Statistics
##
##           Reference
## Prediction lots some very_few
##   lots        2    0        2
##   some        0  263        0
##   very_few    1   10       85
##
## Overall Statistics
##
##                Accuracy : 0.9642
##                  95% CI : (0.9395, 0.9808)
##     No Information Rate : 0.7521
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9086
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: lots Class: some Class: very_few
## Sensitivity             0.666667      0.9634          0.9770
## Specificity             0.994444      1.0000          0.9601
## Pos Pred Value          0.500000      1.0000          0.8854
## Neg Pred Value          0.997214      0.9000          0.9925
## Prevalence              0.008264      0.7521          0.2397
## Detection Rate          0.005510      0.7245          0.2342
```

```
## Detection Prevalence    0.011019      0.7245          0.2645
## Balanced Accuracy       0.830556      0.9817          0.9686
```

6b) Testing k-NN on ride4UT

```
TestResmod1 <- predict(mod1, newdata = ride4UT, type="raw")

confusionMatrix(TestResmod1, ride4UT$complaints)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction lots some very_few
##   lots        0    0        0
##   some        0  262        6
##   very_few    3   11       81
##
## Overall Statistics
##
##                Accuracy : 0.9449
##                  95% CI : (0.9162, 0.966)
##     No Information Rate : 0.7521
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8558
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                         Class: lots Class: some Class: very_few
## Sensitivity                0.000000      0.9597          0.9310
## Specificity                1.000000      0.9333          0.9493
## Pos Pred Value                  NaN      0.9776          0.8526
## Neg Pred Value             0.991736      0.8842          0.9776
## Prevalence                 0.008264      0.7521          0.2397
## Detection Rate             0.000000      0.7218          0.2231
## Detection Prevalence       0.000000      0.7383          0.2617
## Balanced Accuracy          0.500000      0.9465          0.9402
```

Evaluation and discussion

Measure/Method	Accuracy(avg)/C5.0Tree	Accuracy(avg)/K-nearest neighbor
Dataset		
ride4UT	0.9642	0.9449

We can use the confusion matrix to evaluate the results. Total number of errors in the tree classifier are 13, whereas, it increases to 20 when using the instance based classifier on the test dataset. This implies that, the number of misclassifications are more in unseen data when using the instance based classifier.

The table above shows that the accuracy also has been effected when using the instance based classifier on unseen data or testing dataset. By taking into consideration increase in number of errors which is quite significant and the decrease in accuracy we can say the tree classifier performs better on unseen data in our case (Arana 2021).

Task 7: Clustering on ride4U Dataset

##View dataset

```
View (ride4U)

##Pre-processing
##Removing useless attributes month,day,holiday,day_of_week

ride4U<-subset(ride4U,select=-c(month,day,holiday,day_of_week))

View (ride4U)
```

##ride4U data normalised

```
preProcValues <- preProcess(ride4U, method = c("range"))

ride4UNorm <- predict(preProcValues, ride4U)


# checking normalised dataset
head(ride4UNorm, 5)

##           city outlook temperature  humidity feel_humidity     wind       us
es
## 1   Gordontown  cloudy   0.4364332 0.1982559   comfortable   strong 0.53415
72
## 2 Robertburgh  cloudy   0.1642142 0.2864150   comfortable    light 0.17478
99
## 3   Gordontown  cloudy   0.8290171 0.2918654   comfortable     calm 0.87137
91
```

```
## 4 Robertburgh  cloudy   0.1356680 0.3091702   comfortable    light 0.18675
23
## 5  Gordontown  cloudy   0.2321954 0.3176182   comfortable moderate 0.37083
54
##   complaints
## 1       some
## 2   very_few
## 3       some
## 4   very_few
## 5   very_few
```

#From nominal to binary - one-hot encoding

```
set.seed(123)
#binarise nominal attributes - one-hot encoding
binaryVars <- dummyVars(~ ., data = ride4U)
newride4U <- predict(binaryVars, newdata = ride4U)

# check the results
head(newride4U,5)
```

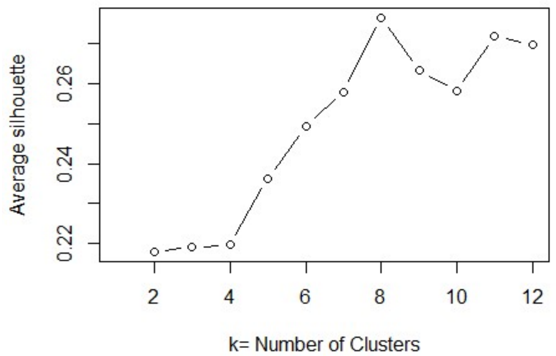
```
##   city.Gordontown city.Robertburgh outlook. outlook.cloudy outlook.rainy
## 1               1                0        0              1             0
## 2               0                1        0              1             0
## 3               1                0        0              1             0
## 4               0                1        0              1             0
## 5               1                0        0              1             0
##   outlook.sunny temperature humidity feel_humidity.comfortable
## 1             0       16.65    38.76                         1
## 2             0        7.40    45.23                         1
## 3             0       29.99    45.63                         1
## 4             0        6.43    46.90                         1
## 5             0        9.71    47.52                         1
##   feel_humidity.dry feel_humidity.humid wind.calm wind.gale wind.light
## 1                 0                   0         0         0          0
## 2                 0                   0         0         0          1
## 3                 0                   0         1         0          0
## 4                 0                   0         0         0          1
## 5                 0                   0         0         0          0
##   wind.moderate wind.strong uses complaints.lots complaints.some
## 1             0           1 5438               0               1
## 2             0           0 1803               0               0
## 3             0           0 8849               0               1
## 4             0           0 1924               0               0
## 5             1           0 3786               0               0
##   complaints.very_few
## 1                   0
## 2                   1
## 3                   0
## 4                   1
## 5                   1
```

#Apply and principal components analysis (PCA).

```
pca_newride4U <- preProcess(newride4U,
                    method = c("pca"))
##pca_newride4U

ride4U2 <- predict(pca_newride4U, newdata = newride4U)

View(ride4U2)
```

#Applying K-means clustering algorithm

```
# For each value of k, k-means is applied. The average silhouette is calculated.
set.seed(123)

sil <-NULL
for (i in 2:12)
{
  res <- kmeans(ride4U2, centers = i, nstart = 25)
  ss <- silhouette(res$cluster, dist(ride4U2))
  sil[i] <- mean(ss[, 3])
}
plot(1:12, sil, type="b", xlab="k= Number of Clusters", ylab="Average silhoue
tte")
```



k=8 seems appears to be best for clustering

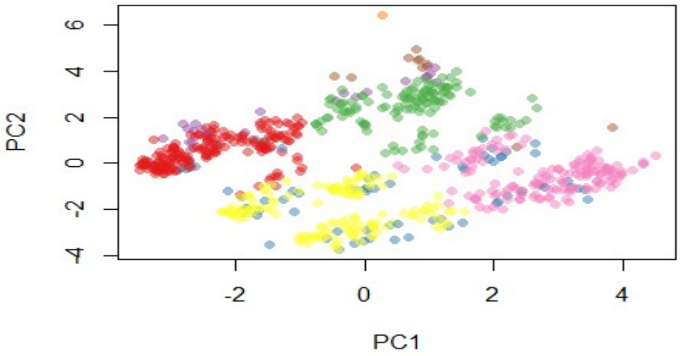
#Apply k-means with k=8

```
set.seed(123)
km <- kmeans(ride4U2, 8, nstart=25, iter.max=1000)
```

#Viewing results

Checking good separation of clusters (and good cohesion) in 2-D.

```
palette(alpha(brewer.pal(9,'Set1'), 0.5))

plot(ride4U2, col=km$clust, pch=16)
```



From the above plot we can say that in a 2D space the instances of the different clusters are mixed within each other.

#Cluster sizes - sort clusters by size

```
sort(table(km$clust))

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


##   5   7   4   2   3   8   6   1
##   1  10  24  71 123 153 160 184
```

The resulting clusters vary in sizes with cluster 1 having the highest number of instances and cluster 5 has the least number of instances.

Discussion

The ideal number of clusters have been achieved using the silhouette method which gives a value of k=8. The algorithm used in this task is K-means. It is a type of batch clustering algorithm. The way in which it works is that it iterates over all instances until convergence It uses a distance metric and partitions instances into disjoint clusters.

- k-means is easy to implement
- It is efficient
- It is easy to explain
- It is non-deterministic, which is also a disadvantage. Using the same value of k obtained, the algorithm can produce different clusters using different centroids. Hence, best results can be obtained by repeating it with different seeds (Arana 2021).

References

- Arana, I.,2021. *Algorithms:Decision Trees*.[online lecture/tutorial/seminar].Data Mining.Robert Gordon University.School of Computing and Digital Media,19/10/2021. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5662104/mod_resource/content/2/02-classificationTrees.pdf [Accessed 15/11/2021].
- T, Neha., 2020. *Data Reduction*. [online]. Place of publication(Unknown): Binary Terms. Available from: https://binaryterms.com/data-reduction.html [Accessed 22/11/2021].
- Gupta, S., 2019. Dealing with Noise Problem in Machine Learning Data-sets: A Systematic Review. *Procedia Computer Science*, Volume(161), Pages 466-474.
- Educate, 2020. What is meant by “noise” in data? What are its sources and how it is affecting results?. [online]. Place of Publication(Unknown): Educate. Available from: https://educatech.in/define-version-space-and-illustrate-it-with-an-example/ [Accessed 22/11/2021].
- Arana, I.,2021. *Evaluation of Learning*.[online lecture/tutorial/seminar].Data Mining.Robert Gordon University.School of Computing and Digital Media,12/10/2021. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5672342/mod_resource/content/4/04-evaluationOfLearning.pdf [Accessed 15/11/2021].
- Arana, I.,2021. *Algorithms:Instance-Based Learning,K-Nearest Neighbour*.[online lecture/tutorial/seminar].Data Mining.Robert Gordon University.School of Computing and Digital Media,28/09/2021. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5662104/mod_resource/content/2/02-classificationTrees.pdf [Accessed 15/11/2021].
- Arana, I., 2021. *Clustering*. [online lecture/tutorial/seminar]. Data Mining. Robert Gordon University. School of Computing Science and Digital Media, 26/10/2021. Available from: https://campusmoodle.rgu.ac.uk/pluginfile.php/5680472/mod_resource/content/6/06-clustering-methods.pdf [Accessed 30/11/2021].