SCHOOL OF COMPUTER SCIENCES
UNIVERSITI SAINS MALAYSIA

CPM 351 : Principles of Data Analytics
Semester I, Academic Session : 2021/2022

Project
Group 22

Project Supervisor: Dr. Wong Li Pei

Names	Matric No.	USM Email Address	Core/Minor
Ng Wei Yi	147630	ngweiyi99@student.usm.my	Minor
Lim Zhong Han	147241	limzhonghan@student.usm.my	Minor
Yeoh Jocelyn	154696	bluieeeejocelyn@student.usm.my	Minor

Date of Submission

31/01/2022

Problem Statement

Based on the dataset from the United Kingdom's government website https://data.europa.eu/data/datasets/road-traffic-accidents/?locale=en, Leeds City Council collects the road traffic accidents across Leeds from the year 2019 with data including accident dates, road classes, number of vehicles involved, road surface, lighting conditions, weather conditions, type of vehicles and severity of any casualties. The Eastings and Northings are generated at the roadside where the accident occurred.

Every year, more than 1.17 million people are killed in road crashes around the world. Deaths and injuries in road traffic accidents pose a serious threat to the nation and have a negative impact on social and economic progress.

Therefore, this project is an analysis to predict the severity of casualties with respect to the road surface, lighting conditions and weather conditions in Leeds. To identify the patterns of how these serious accidents happened and the key factors, two suitable machine learning models are performed and the most accurate machine learning model will be chosen to reduce the number of accidents in the future.

Objective

Firstly, the objective of this project is to recognize certain key factors that affect the severity of casualties of car accidents. Secondly, the objective is to choose the best machine learning model to build a champion model to predict the severity of casualties of car accident victims in the UK. The champion model should be able to accurately predict the severity of casualties of car accident victims without any detailed information such as lighting conditions and road surface in real-time or in the future.

Data Preprocessing

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format which will enhance the performance of the machine learning algorithms chosen to be used. The dataset from the United Kingdom's government website https://data.europa.eu/data/datasets/road-traffic-accidents/?locale=en will undergo the data preprocessing process by using certain libraries and functions found in R.

Firstly, libraries are imported to R to be used in data preprocessing, some libraries shown below are for the machine learning models chosen

```
library(ggplot2)
library(dplyr)
library(scatterplot3d)
library(klaR)
library(caret)
library(lattice)
library(rpart)
library(rpart.plot)
library(data.tree)
library(caTools)
```

Set the working directory then read the CSV files using *read.csv()* function and the internal structure of the dataset is displayed using *str()* function. Summary of the imported data is shown below

```
> summary(road_accident)
 Reference.Number   Grid.Ref..Easting Grid.Ref..Northing Number.of.Vehicles Accident.Date
 Length:1907        Min.   :414747    Min.   :423157     Min.   :1.000      Length:1907
 Class :character   1st Qu.:427396    1st Qu.:431198     1st Qu.:2.000      Class :character
 Mode  :character   Median :430096    Median :433982     Median :2.000      Mode  :character
                    Mean   :430274    Mean   :434037     Mean   :1.969
                    3rd Qu.:432241    3rd Qu.:435947     3rd Qu.:2.000
                    Max.   :445481    Max.   :449599     Max.   :6.000
  Time..24hr.   X1st.Road.Class X1st.Road.Class...No  Road.Surface   Lighting.Conditions
 Min.   :   0   Min.   :1.0     Length:1907          Min.   :1.000   Min.   :1.000
 1st Qu.:1000   1st Qu.:3.0     Class :character     1st Qu.:1.000   1st Qu.:1.000
 Median :1500   Median :6.0     Mode  :character     Median :1.000   Median :1.000
 Mean   :1385   Mean   :4.4                          Mean   :1.322   Mean   :1.912
 3rd Qu.:1756   3rd Qu.:6.0                          3rd Qu.:2.000   3rd Qu.:4.000
 Max.   :2355   Max.   :6.0                          Max.   :9.000   Max.   :7.000
 Weather.Conditions Local.Authority    Vehicle.Number  Type.of.Vehicle  Casualty.Class  Casualty.Severity
 Min.   :1.000      Length:1907        Min.   :1.000   Min.   : 1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:1.000      Class :character   1st Qu.:1.000   1st Qu.: 9.000   1st Qu.:1.000   1st Qu.:3.000
 Median :1.000      Mode  :character   Median :2.000   Median : 9.000   Median :1.000   Median :3.000
 Mean   :1.242                         Mean   :1.558   Mean   : 8.345   Mean   :1.567   Mean   :2.802
 3rd Qu.:1.000                         3rd Qu.:2.000   3rd Qu.: 9.000   3rd Qu.:2.000   3rd Qu.:3.000
 Max.   :9.000                         Max.   :5.000   Max.   :97.000   Max.   :3.000   Max.   :3.000
 Sex.of.Casualty Age.of.Casualty
 Min.   :1.000   Min.   : 1.0
 1st Qu.:1.000   1st Qu.:22.5
 Median :1.000   Median :32.0
 Mean   :1.397   Mean   :35.8
 3rd Qu.:2.000   3rd Qu.:47.0
 Max.   :2.000   Max.   :95.0
```

Convert the attributes of Road.Surface, Lighting.Conditions, Weather.Conditions and Casualty.Severity to factor using *as.factor()* function and by using *dataframe[, c()]* function to select specific columns by names, the new summary of the imported data is shown below

```
> summary(road_accident_2)
 Road.Surface Lighting.Conditions Weather.Conditions Casualty.Severity
 1:1359       1:1389              1      :1633       1:  22
 2: 521       4: 436              2      : 215       2: 334
 3:   3       5:   9              5      :  25       3:1551
 4:  17       6:  43              4      :  19
 5:   5       7:  30              8      :   8
 9:   2                           3      :   3
                                  (Other):   4
```

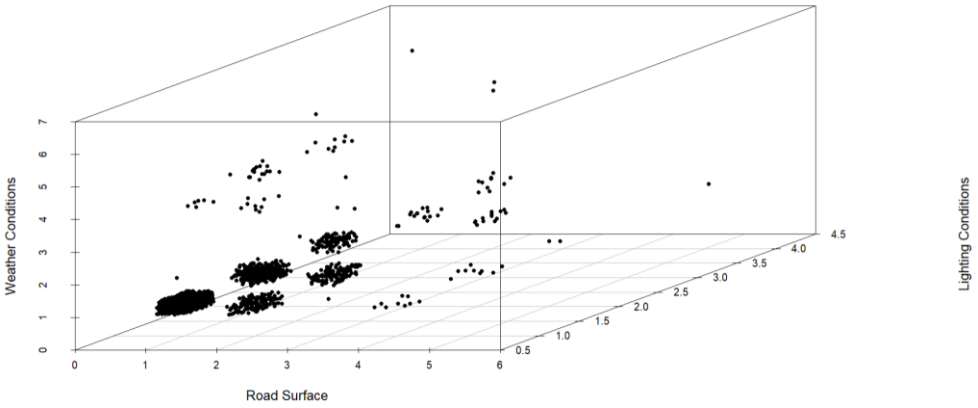
Then, by using *sum(is.na())* function to determine whether there are null values in Road.Surface, Lighting.Conditions, Weather.Conditions and Casualty.Severity, the output is shown below

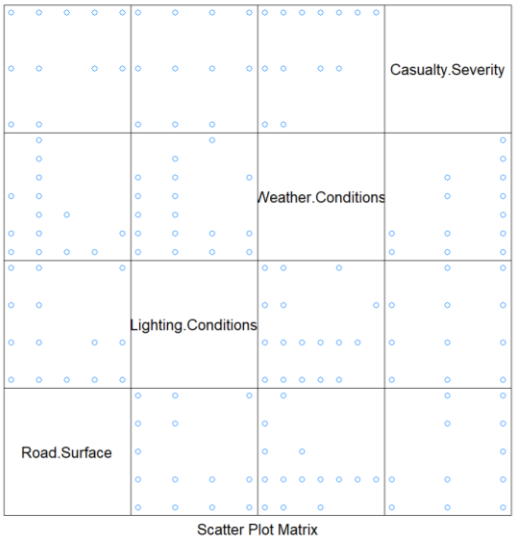
```
> sum(is.na(road_accident_2$Road.Surface))
[1] 0
> #missing value in road_accident_2$Lighting.Conditions
> sum(is.na(road_accident_2$Lighting.Conditions))
[1] 0
> #missing value in road_accident_2$Weather.Conditions
> sum(is.na(road_accident_2$Weather.Conditions))
[1] 0
> #missing value in road_accident_2$Casualty.Severity
> sum(is.na(road_accident_2$Casualty.Severity))
[1] 0
```

Using *subset()* function to remove the outliers, *Weather.Conditions == 8* and *Weather.Conditions == 9* from Weather.Conditions and *Lighting.Conditions == 3* and *Lighting.Conditions == 7* from Lighting.Conditions because based on the data description of this dataset, these conditions are either unknown conditions or other. The new summary is shown below

```
> summary(road_accident_3)
 Road.Surface Lighting.Conditions Weather.Conditions Casualty.Severity
 1:1330       1:1384              1      :1603       1:  21
 2: 514       4: 433              2      : 215       2: 327
 3:   2       5:   9              5      :  25       3:1519
 4:  16       6:  41              4      :  19
 5:   5       7:   0              3      :   3
 9:   0                           6      :   1
                                  (Other):   1
```

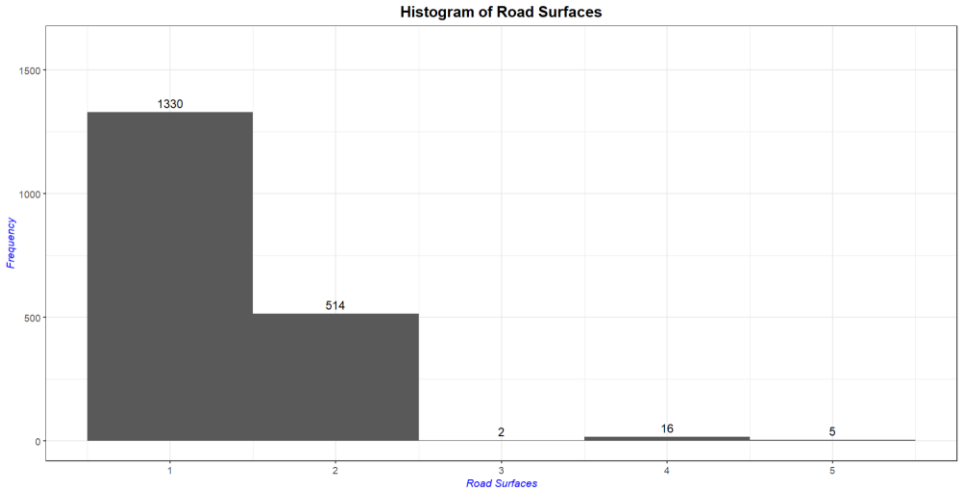
Convert the attributes of Road.Surface, Lighting.Conditions, Weather.Conditions and Casualty.Severity to numeric in order to plot the 3D scatterplot and the scatterplot matrix. *scatterplot3d()* function is used to plot the 3D scatterplot while *splom()* function is used to plot the scatterplot matrix. Both plots are shown below

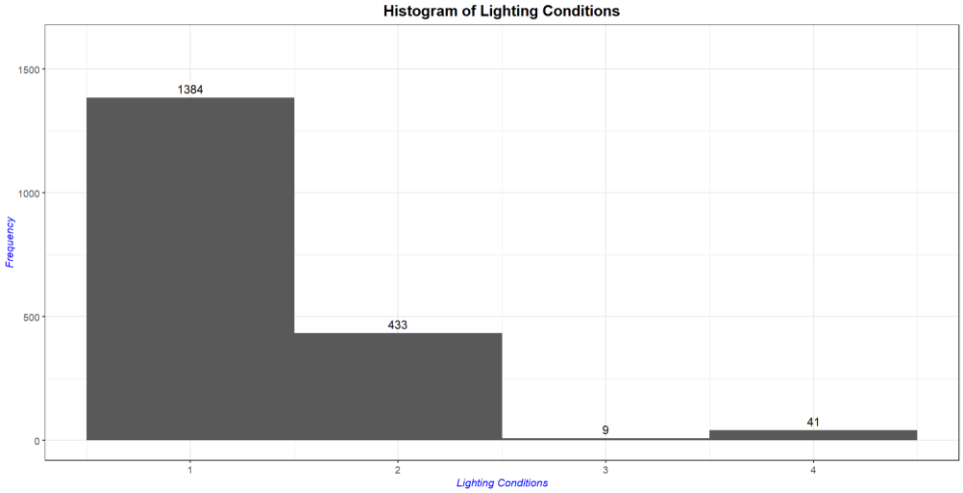


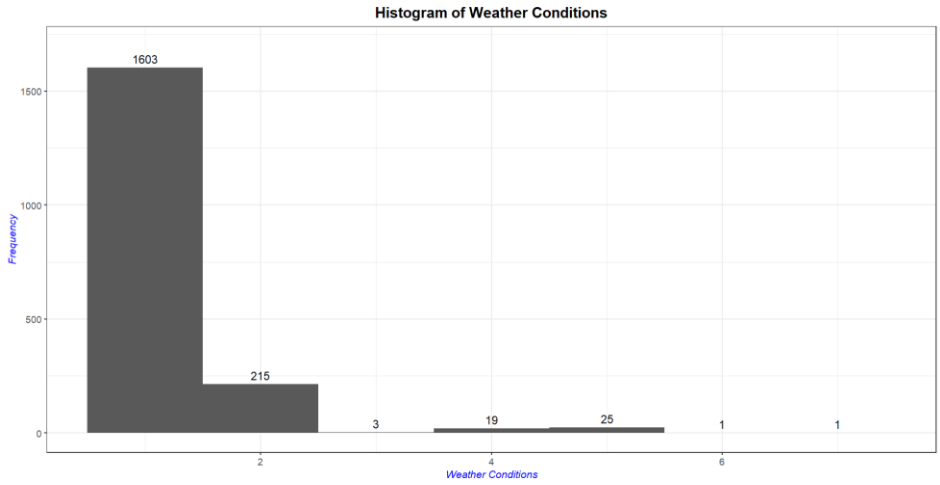


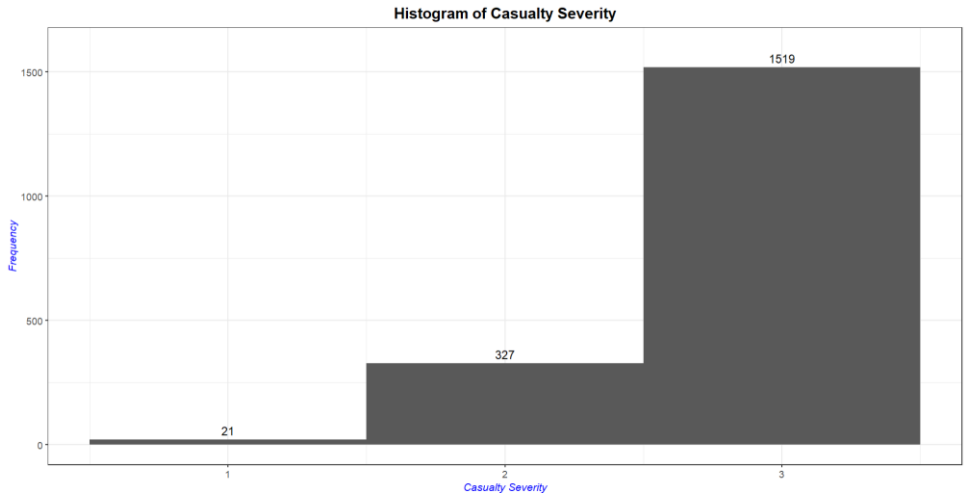
Scatter Plot Matrix

Using *ggplot2* library to plot the histograms of Road.Surface, Lighting.Conditions, Weather.Conditions and Casualty.Severity. The histograms are shown below









Convert the attributes of Road.Surface, Lighting.Conditions, Weather.Conditions and Casualty.Severity back to factor using *as.factor()* function for the next step, applying the chosen machine learning algorithms, K-modes Clustering and Decision Tree Classifier on the cleaned dataset.

Machine Learning Models

Two machine learning algorithms are employed in the data analysis which are unsupervised K-modes clustering and supervised decision tree classification model.

K-modes Clustering

K-modes clustering is one of the unsupervised machine learning algorithms that is used to cluster categorical variables. It uses the dissimilarities between the data points. The lesser the dissimilarities, the more similar the data points are. It uses modes instead of means where mode is the value that appears most often in a set of data values.

Packages such as *"klaR"* and *"caret"* are installed. The *"klaR"* package is used to perform K-modes clustering algorithm while *"caret"* package will be used to generate the confusion matrix for model evaluation. The algorithm is first operated under a 'for' loop with a range of 10 data values to obtain a fairly good and precise result. This is due to in K-modes clustering, different centroids will be chosen which will result in inconsistency. Before performing the algorithm, set seed first so that the sampling data is reproducible.

An empty vector named *accuracy_score* is created to store the values of accuracy from the confusion matrix. Then, apply K-modes clustering and view the results using *kmodes()* function with arguments including *data* that represents the chosen dataset, *modes* of a set of the distinct cluster modes, *iter.max = 10* means the maximum number of iterations allowed which is 10, *weighted = FALSE* means the weighted version of the distance which is not applied here, and *fast = TRUE* which means a fast version of the algorithm should be applied. The following code for this function is used in the variable named *k_modes* as shown below:

```
>   #KModes CLustering
>   k_modes = kmodes(road_accident_3, 3, iter.max = 10, weighted = FALSE, fast = TRUE)
>   k_modes
K-modes clustering with 3 clusters of sizes 454, 473, 940

Cluster modes:
  Road.Surface Lighting.Conditions Weather.Conditions Casualty.Severity
1            1                   2                  1                 2
2            2                   1                  1                 3
3            1                   1                  1                 3

Clustering vector:
   [1] 3 3 3 3 3 3 3 3 3 1 1 1 3 1 3 3 3 3 3 3 1 3 3 2 1 3 3 3 3 3 3 3 1 1 1 3 3 3 3 2 1 1 1 3 1 1 1 1 2 1 2 2 3 1 1 1
  [56] 2 3 3 3 1 1 1 1 1 2 3 3 1 1 3 1 2 2 1 3 2 1 2 2 1 1 2 3 3 2 2 1 2 1 3 3 3 3 2 3 3 3 3 2 2 3 3 1 1 2 2 2 2 2 2
 [111] 2 1 2 2 2 3 2 1 3 3 2 3 1 1 2 2 2 3 1 2 2 2 2 2 2 1 3 1 2 3 3 3 1 1 1 3 3 2 2 1 1 3 3 3 3 3 2 2 2 2 3 3 2 1 2 3 3
 [166] 3 3 3 3 3 3 3 2 1 1 2 1 3 3 3 3 3 3 2 1 1 1 2 3 3 2 2 1 2 2 1 3 2 2 2 2 2 1 1 1 1 1 3 1 3 3 3 2 2 2 2 3 1 2 2 3 2
 [221] 3 3 1 3 1 1 1 3 3 1 3 3 1 1 3 1 3 3 3 3 3 3 3 1 1 3 1 3 1 3 3 2 1 1 3 3 3 3 3 1 3 3 1 1 1 3 3 1 1 3 3 1 3 3 1 1 3 3
 [276] 1 1 3 1 3 3 3 3 1 3 3 1 1 3 3 3 1 3 1 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 1 1 1 3 3 2 2 1 2 1 2 3 2 2 1 3 1 3 3 1 3 3 1
 [331] 2 1 3 3 3 3 3 1 3 3 3 3 3 2 2 1 1 2 3 3 2 3 3 3 2 2 2 2 2 2 3 3 3 3 3 3 3 1 1 1 1 2 2 3 1 3 2 2 3 2 2 2 2 2 3 1 2
 [386] 3 3 3 3 2 3 3 3 2 2 3 2 3 3 3 3 3 1 3 3 3 3 3 3 3 1 3 1 1 3 3 3 3 1 1 1 1 3 3 3 3 1 1 3 3 3 3 3 3 1 3 3 3 3 3 1 1 1
 [441] 1 1 3 3 3 1 1 1 3 1 1 1 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3 3 3 1 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 2 2 3 3 1
 [496] 3 3 1 1 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 1 3
 [551] 3 3 1 1 1 3 1 3 1 3 3 1 1 1 1 3 3 3 3 3 3 3 1 1 3 3 1 1 1 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3 2 3 1
 [606] 3 2 3 3 1 3 1 1 1 1 3 3 3 3 3 3 3 3 3 3 1 3 3 2 3 3 3 3 3 3 3 1 1 3 3 3 3 3 3 3 2 1 2 2 2 2 3 3 3 2 2 2 3 2 2 2 2 2
 [661] 2 2 2 2 2 2 2 1 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 3 1 1 3 3 1 1 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1
 [716] 3 3 1 3 3 3 3 3 1 3 3 2 3 2 3 2 2 3 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 3 3 3 2 2 2 3 1 3 3 3 3 3
 [771] 1 1 1 2 2 1 1 3 3 2 3 3 3 1 2 1 3 2 2 1 2 2 2 2 2 2 2 3 3 2 2 2 2 3 3 2 2 1 3 3 3 3 3 3 3 3 1 2 2 2 2 3 1 3 3
 [826] 3 3 3 3 3 3 1 3 3 3 3 3 3 2 3 3 3 3 3 3 3 1 1 3 3 3 3 2 2 3 1 3 3 3 2 2 2 2 2 2 2 2 3 3 2 3 1 3 3 3 3 3 3 3 1 3 1 3
 [881] 3 1 3 3 3 3 3 1 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1
 [936] 3 3 3 1 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 1 3 3 1 1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3
 [991] 1 3 3 2 2 2 2 2 2 1
 [ reached getOption("max.print") -- omitted 867 entries ]

Within cluster simple-matching distance by cluster:
[1] 459 483  29

Available components:
[1] "cluster"    "size"       "modes"      "withindiff" "iterations" "weighted"
```

A new table is created under *kmodes_table* with *cbind()* function to combine *road_accident_3* dataset followed by *cluster* column from *k_modes* dataset:

```
>   kmodes_table = cbind(road_accident_3, k_modes$cluster)
>   kmodes_table
  Road.Surface Lighting.Conditions Weather.Conditions Casualty.Severity k_modes$cluster
1            1                   1                  1                 3               3
2            1                   1                  1                 3               3
3            1                   1                  1                 3               3
4            1                   1                  1                 3               3
5            1                   1                  1                 3               3
6            1                   1                  1                 3               3
```

Next, all the variable names from the table formed are renamed by using *colnames()[i]* where *i* is the index of the columns table. For example, *colnames(kmodes_table)[5] <- "Predicted Casualty Severity"* where "cluster" is changed to "Predicted Casualty Severity".

Moreover, the attributes from *k_modes_table* such as Road Surface, Lighting Conditions, Weather Conditions, Casualty Severity and Predicted Casualty Severity are cast to factor using *as.factor()* function in order to perform confusion matrix.

Before generating the confusion matrix, a table named *xtab* is created with the prediction and actual values obtained from *k_modes_table*.

```
>   xtab <- table(kmodes_table$`Predicted Casualty Severity` , kmodes_table$`Casualty Severity`)
>   xtab

     1   2   3
  1   9 275 170
  2   5  52 416
  3   7   0 933
```

Confusion matrix is then generated on *xtab*.

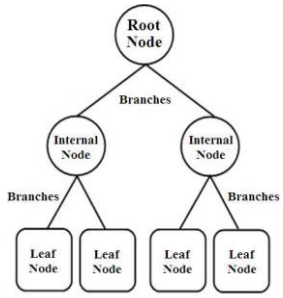
```
>   con_mat <- confusionMatrix(xtab)
>   con_mat
Confusion Matrix and Statistics


     1   2   3
  1   9 275 170
  2   5  52 416
  3   7   0 933

Overall Statistics

               Accuracy : 0.5324
                 95% CI : (0.5095, 0.5552)
    No Information Rate : 0.8136
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1393

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:
```

Only the 'Accuracy' value obtained from the Overall Statistics of the confusion matrix is extracted and stored under *score*. Lastly, 10 accuracy values from the data are appended in the vector function *accuracy_score* and shown below:

```
>   #Extract accuracy score
>   score <- con_mat$overall['Accuracy']
>   accuracy_score <- append(accuracy_score, score)
> #Show the accuracy scores obtained from KModes CLustering
> accuracy_score
   Accuracy    Accuracy    Accuracy    Accuracy    Accuracy    Accuracy    Accuracy    Accuracy    Accuracy
0.650241028 0.532404928 0.007498661 0.525441885 0.140332084 0.525441885 0.003749330 0.096411355 0.046063203
   Accuracy    Accuracy
0.189073380 0.189073380
```

Decision Tree Classifier

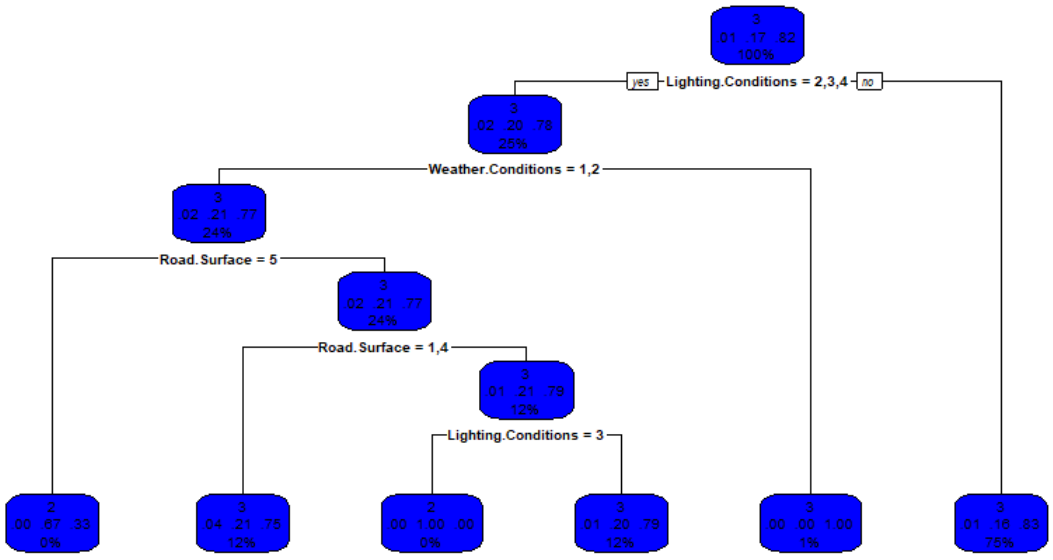
A decision tree is a map of the possible outcomes of a series of related choices. It is a flowchart-like structure where each internal node is a point where a choice must be made. Each branch extending out from the node represents an alternative outcome that is mutually exclusive based on the given data. The decision tree will keep on branching off for each criteria and the final node at the last branch of the tree is called the leaf. Each leaf is considered as a class.



(a)

rpart library is used to perform the decision tree algorithm. The decision tree algorithm is performed under a 'for' loop to have different results for the train-test split. The seed is set to be a specific value for each loop and consistent with the previous k-modes clustering algorithm so that the train and test datasets will be the same for better comparison and the results will be reproducible.

An empty vector named *accuracy_score_2* is created to store the values of accuracy from the confusion matrix. Then, split the data into train and test datasets with a ratio of 8:2 then train decision tree algorithm, *rpart()*, on the train dataset with gini index as the attribute selection. There is not much difference between using gini index and information gain as parameter in attribute selection for decision tree, however gini index will have less computation cost . The decision tree obtained is shown below



Then, predict the Casualty.severity attribute for the test dataset using the decision tree model. Below is the confusion matrix for one of the seeds.

```
pred   1   2   3
   1   0   0   0
   2   0   0   1
   3   5  71 294

Overall Statistics

               Accuracy : 0.7925
                 95% CI : (0.7476, 0.8326)
    No Information Rate : 0.7951
    P-Value [Acc > NIR] : 0.5813

                  Kappa : -0.0052

 Mcnemar's Test P-Value : NA
```

The 'Accuracy' value obtained from the Overall Statistics of the confusion matrix is extracted and stored under *score*. Lastly, 10 accuracy values from the data are appended in the vector function *accuracy_score_2* and shown below:

```
> accuracy_score_2
 Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy  Accuracy
0.8381201 0.8252427 0.8241470 0.8286517 0.8067010 0.7973333 0.8193717 0.8392371 0.8091168 0.7924528
```

Hypothesis Testing

H_0 (Null Hypothesis): There is no significant difference between the mean of accuracy of the K-modes Clustering model and the accuracy of the Decision Tree Classifier model.

H_1 (Alternative Hypothesis): There is a significant difference between the mean of accuracy of the K-modes Clustering model and the accuracy of the Decision Tree Classifier model.

The Welch's T-test is chosen as it can perform tests on data with unequal variance. It is used to compare the mean of two populations. It is performed using the *t.test()* function with *var.equal = FALSE* to imply that the variance is unequal. The results are shown below.

```
        Welch Two Sample t-test

data:  accuracy_score and accuracy_score_2
t = -6.7498, df = 9.0719, p-value = 8.056e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.7292627 -0.3634806
sample estimates:
mean of x mean of y
0.2716658 0.8180374
```

The p-value obtained from Welch's t-test is less than 0.05. Thus, the null hypothesis is rejected. There is a significant difference between the mean of accuracy of the K-modes Clustering model and the accuracy of the Decision Tree Classifier model. Since "t" has a negative value, the accuracy for the K-modes clustering model is less than the Decision Tree Classifier model.

Conclusion

In conclusion, the classifier model performs better than the clustering model for this dataset. Clustering performs poorly for discrete data. The results are also inconsistent, as seen from the accuracy score of K-means clustering where the range for the accuracy score is very wide. The decision tree classification model on the other hand, can accurately predict the casualty severity level based on the three chosen attributes. It has an average accuracy of 81.8% when predicting the casualty severity level for the respective test datasets.

Problems and Pitfalls

Initially, when we tried to visualize the Decision Tree Classifier using *rpart.plot()* function, there is only one node that appeared in the output. Hence, the control parameters must be lowered for the decision tree because the default parameters are too high so it does not split.

Moreover, we applied the Hypothesis Testing on the wrong variable, which is our data although the variable we want to do testing on is the mean accuracies of two different machine learning models. Eventually, we realised our mistake and applied the Hypothesis Testing correctly on the mean accuracies of two different machine learning models.

Furthermore, we spent some time on deciding which Hypothesis Test should be used in this project as there are many different tests such as ANOVA test, student’s t-test and Welch's t-test. Eventually, we settled on Welch's t-test as it is the most suitable Hypothesis Test for this project.

Despite all the hard times and failures we came across, the team spirit embedded in our heart throughout the project has got us over it with the help of everyone in the team who plays their role perfectly and responsibly. We brainstormed over the issues together and managed to get a relatively desired output eventually.