UK Traffic Accidents:
A Spatial and Machine Learning Approach

by

Issam Jaber (1740856), Olanike Ikumelo (1626595), Haoyu Li (1726320) and Qi Ding (1702658)

Learning Development Project

Brunel University

April 2018

Table of Contents

Chapter 1 Introduction

Fear of being in a traffic accident is a major worry for many. Those with driving phobia, find it difficult to adjust in a modern world that revolves around travel. Many people in the UK share this concern and it's not surprising with the RAC revealing that "There were 24,101 people seriously injured in reported road traffic accidents in 2016", (RAC, 2018).

In our paper we attempt to aid those worried about their safety when travelling by road, we attempt to find the specific conditions that result in dangerous accidents as well as identifying areas where dangerous accidents occur more frequently. The outcome of the report could also benefit the UK government, Transport for London, Environmental Health Association and car manufacturers. Various applications were utilized in carrying out this research. We will use data surrounding traffic accidents in the UK, dating from 2005-2015.

In order to find the dangerous conditions, we use several machine learning methods such as: Random Forest, K-nearest neighbours, Multiple Linear Regression, Decision Tree, Neural Networks and Logistic Regression. We will answer questions such as: "Is it safe to travel in the snow?" and "Is it advisable to travel in the dark?".

Spatial analysis will also be used to discover the areas where dangerous accidents occur more frequently. We will use a choropleth map to visualise accident frequency across UK districts. An interactive cluster map will also be created, where each accident is marked on a map. This map along with the choropleth will be used to create an app for users to be able to search their location or where they want to visit. The can then learn if there were dangerous accidents that happened nearby. This will hopefully provide peace of mind for those worried about travelling to a new area in the UK.

In this paper we will begin by describing the dataset. In chapter 2 we will review relevant literature. Chapter 3 will explore the data using tableau, in chapter 4 we will share the methods and results of the spatial analysis. Chapter 5 will reveal the results obtained from the various machine learning models. In the final chapter a critical evaluation will be conducted for the project as a whole.

1.1 The Dataset

The objective of this project was to discover the most dangerous areas and conditions to drive in, thus, a dataset of UK Car Accidents for the years 2005-2015 was used, (Kaggle.com, 2018). The dataset consisted of 32 variables and 1,780,653 observations, most of the categorical variables were pre-formatted to levels. In order for us to understand the levels, a guide for the dataset was also downloaded. Table 1 describes the main variables in the dataset which we used for our project.

Variable	Description	Type
Longitude	Coordinate of the accident	num
Latitude	Coordinate of the accident	num
Accident_Severity	Severity of the Accident, where 1=Fatal, 2=Serious and 3=Slight	int
Number_of_Casualties	Total number of casualties per accident	int
Date	Full date of the accident	chr
Day_of_Week	The day of the week in which the accident occurred, numbered from 1-7, starting with Monday	int
Time	The time in which the accident occurred	chr
Local_Authority_(District)	The district in which the accident occurred, each district is numbered in the dataset.	int
Light_Conditions	The level of light during the accident. 1=Daylight, 4=Darkness-lights lit, 5=Darkeness-lights unlit, 6=Darkness-no lighting, 7=Darkness lighting unknown and -1=Data missing	int
Weather_Conditions	The weather condition at the time of the accident. 1=Fine no high winds, 2=Raining no high winds, 3=snowing no high winds, 4= fine with high winds, 5=Raining with high winds, 6=snowing with high winds, 7=Fog or Mist. 8=other, 9=Unknown and -1=data missing	int
Road_Surface_Conditions	The condition of the road at the time of the accident. 1=Dry, 2=Wet or damp, 3=Snow, 4=Frost or ice, 5=Flood over 3cm, 6=Oil or diesel, 7=Mud and -1=data missing	int
Urban_or_Rural_Area	Whether the area the accident occurred was an Urban area or Rural area, 1=Urban, 2=Rural and 3=Unallocated	int

Table 1

Chapter 2 Literature Review

Abdalla et al. (1997), examined the connection regarding rate of casualties and how far the accidents were from the areas of residence. The result was unsurprising, the rate of casualties was higher nearer to the areas of residence. This may be due to there being a greater deal of exposure. The study uncovered that the frequency of casualties for those from poor areas was significantly higher than those from more affluent areas. This could be studied further in terms of urban and rural areas in our case.

Abdelwahab and Abdel-Aty (2001), examined traffic accident which occurred in Central Florida in 1997. The study split injury severity into three categories: disabling injury, possible injury and no injury. The analysis contrasted the performance of Multi-layered Perceptron (MLP) and Fuzzy Artmap (FA), and found the former had a higher level of accuracy. The MLP received a classification accuracy of 66%, whereas, the FA received 56%.

Bedard et al. (2002), used a multivariate logistic regression to establish the main influence of a driver's actions on the risk of fatalities. They discovered that by increasing seatbelt use, decreasing average speed are likely to aid in avoiding fatalities.

Richard and Ray (2017), analysed road accident data for two cities, Fredericton and Laval. Spatial frameworks were used to present a comparative analysis of accidents between the cities. They used random forest classification to predict casualties from traffic accidents. Their model found that the type of accident was most influential when determining whether an accident was serious or fatal.

Fred Mannering (2018) provided numerous examples from a variety of fields that indicate that there is strong behavioural evidence to suggest that temporal instability is likely an important issue in contemporary analyses of accident data. They found the temporal elements associated with individual behaviour and the aggregate trends that result from these (like long-term decline fatalities per mile and the phenomenon of aggregate economics) are important factors to consider when developing modelling approaches and interpreting model findings.

Chenhui Liu and Anuj Sharma (2017) analysed the crash frequency in American car accidents through mathematical modelling and visualization methods by generating choropleth maps. They found the impact of a smaller time scale, such as season or month, should be explored, as this may offer more details about crash frequency changing trends and show the influences of periodic factors such as weather.

Xiaoxiang Ma and Suren Chen (2016) investigated the application of multivariate space-time models to jointly analyse crash frequency by injury severity levels in fine temporal scale. A multivariate space-time modelling framework was proposed within Full Bayesian paradigm which focuses on finding best specifications for spatial and temporal random effects.

Chapter 3 Data Exploration

In order to better understand our data, visualisations have been produced using Tableau.

Figure 1 shows the total number of casualties that occurred for each level of accident severity. From the figure we can clearly see that the overwhelming majority of accidents were slight accidents, with 2 million casualties in the ten years from 2005-2015. Serious and Fatal accidents accounted for a much smaller proportion of casualties, with 350,000 and 40,000 casualties, respectively. Therefore, we can conclude that less severe accidents are more likely to occur.

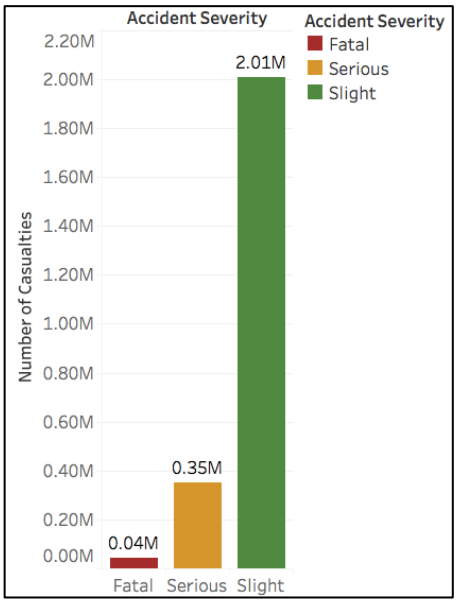

Figure 1

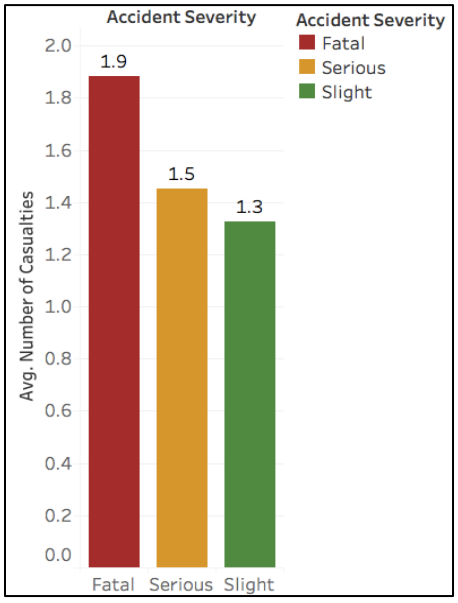

Figure 2

Figure 2, follows a similar premise as Figure 1, however, the average number of casualties is calculated for each level of severity. The figure shows that Fatal accidents had the highest number of casualties on average, with 1.9 casualties per accident. Serious and Slight accidents had less casualties on average with 1.5 and 1.3, respectively. Thus, accident severity and number of casualties are positively correlated, a more severe accident will result in a greater number of casualties on average.

Figure 3, displays number of casualties of time. The figure, shows that the number of casualties has fallen gradually since 2005, from 270,000 in 2005 to just less than 200,000 in 2014. The forecast also predicts that the number of casualties will continue to fall in the following three years to 169,000 in 2017.

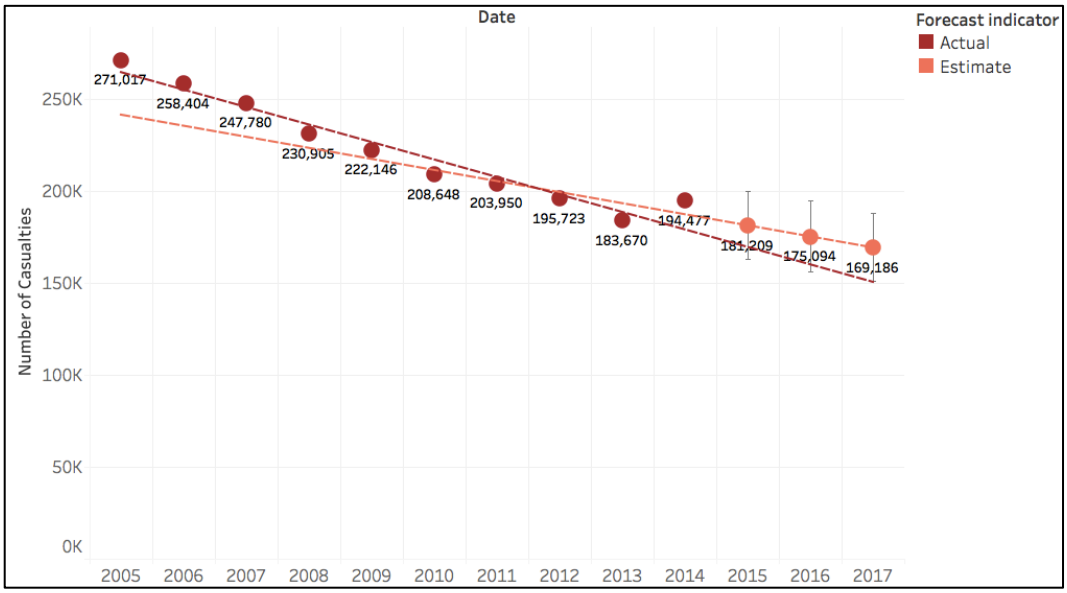

Figure 1

Figures 4 through to 8 have been filtered for serious and fatal accidents in terms of severity, this is to find the most dangerous conditions. Figure 4 looks at the average number of casualties on each day of the week in order to find the most dangerous days to travel. From the chart we deduce that weekends are more dangerous than on weekdays, with Sunday being the most dangerous day to travel.

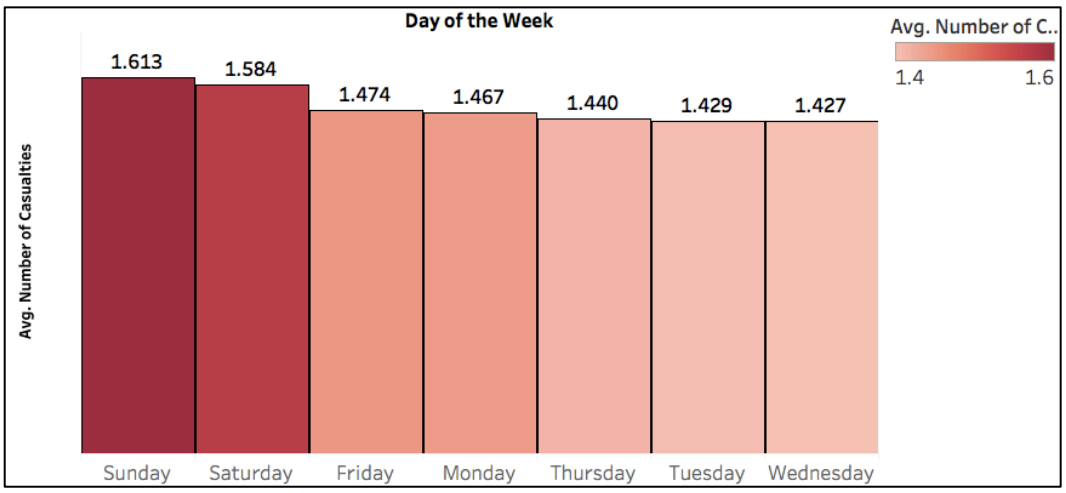

Figure 2

Figure 5 looks at the average number of casualties for each level of lighting. From the chart we can see that more accidents occur on average when there's no light.

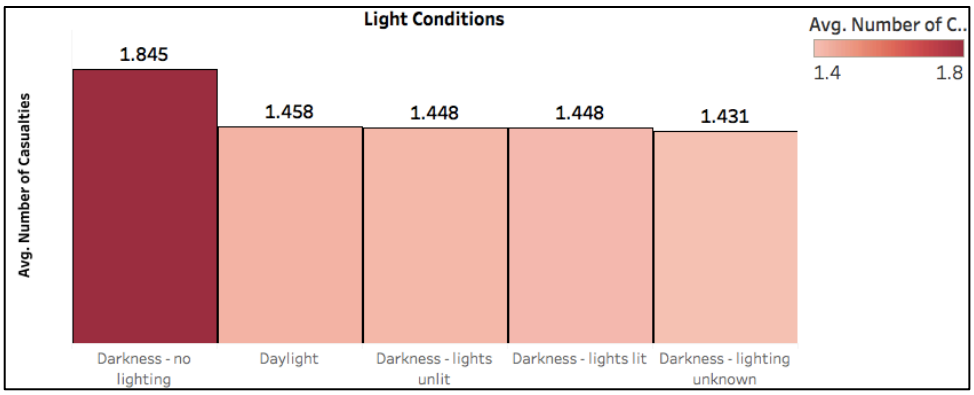

Figure 3

Figure 6 looks at the average number of casualties for different road conditions. From the chart we can see that more accidents occur when there's flood with almost 2 casualties per accident.

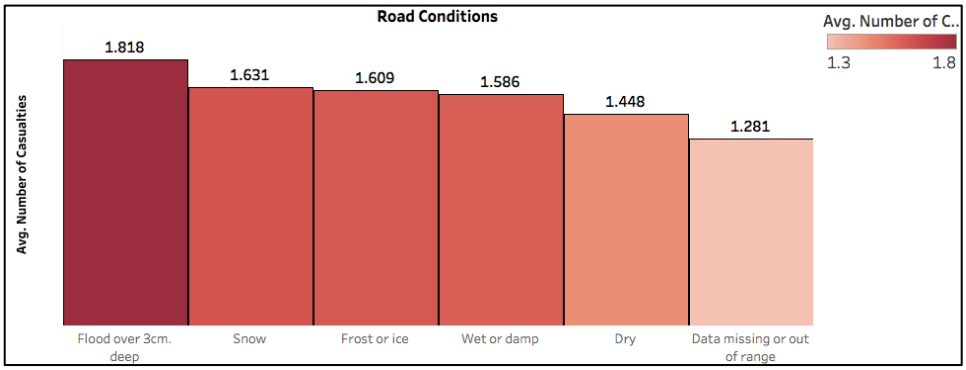

Figure 4

Figure 7 looks at the average number of casualties for different weather conditions. From the chart we can see that more accidents occur during seemingly unpleasant weather. Driving is most dangerous during snow and high winds with an average of 1.896 casualties per accident, foggy weather comes close with 1.805.

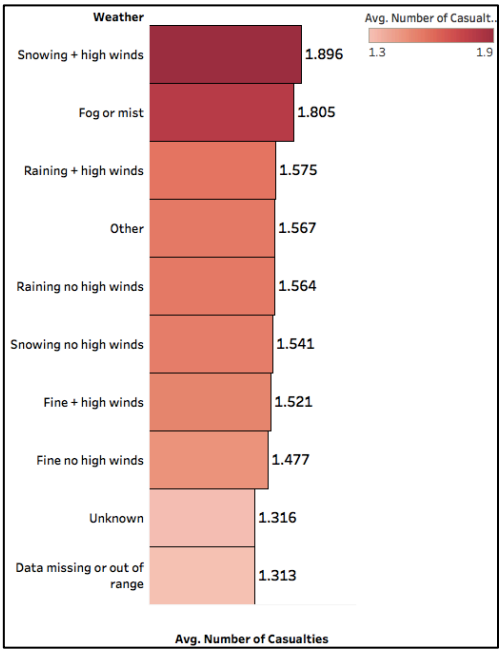

Figure 7

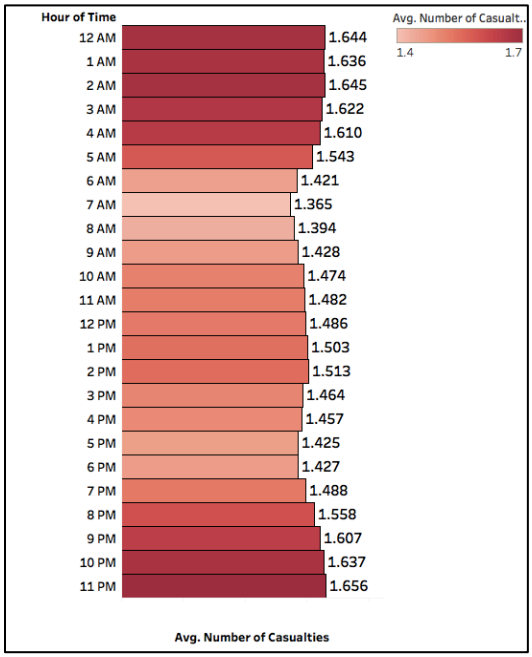

Figure 8

Figure 8 shows the average number of casualties for different times of the day. From the chart we can see that the highest number of casualties on average occur during hours of low light, between 8pm until 5am.

Chapter 4 Spatial Analysis

4.1 Introduction

Since we are dealing with geospatial data, it's wise to conduct a spatial analysis to uncover some meaningful information. We will do this in two parts, first a choropleth map will be created, this will aid us in discovering where accidents are likely to occur. Then, we will create an interactive cluster map where users can search a chosen location and see the accidents that occurred near that location in the past.

Since, including the entire dataset will result in a slow and frustrating visualisation, we cleaned the data in a manner which would result in efficient maps containing the most meaningful information.

4.2 Data Cleaning

We started off by finding out what information was necessary to keep, since reducing the size of the dataset was vital to the overall success of this task. With too much data, the maps would fail to generate and crash mid-process. The first step was to find and remove any NA values in the dataset.

```
> df_NA_count <- apply(is.na(df), 2, sum)
> df_NA_count
                          ï..Accident_Index                      Location_Easting_OSGR
                                          0                                       138
                      Location_Northing_OSGR                                  Longitude
                                        138                                       138
                                   Latitude                             Police_Force
                                        138                                         0
                           Accident_Severity                        Number_of_Vehicles
                                          0                                         0
                       Number_of_Casualties                                      Date
                                          0                                         0
                                Day_of_Week                                      Time
                                          0                                       151
                 Local_Authority_.District.                   Local_Authority_.Highway.
                                          0                                         0
                            X1st_Road_Class                          X1st_Road_Number
                                          0                                         0
                                  Road_Type                              Speed_limit
                                          0                                         0
                           Junction_Detail                          Junction_Control
                                          0                                         0
                           X2nd_Road_Class                          X2nd_Road_Number
                                          0                                         0
          Pedestrian_Crossing.Human_Control     Pedestrian_Crossing.Physical_Facilities
                                          0                                         0
                          Light_Conditions                         Weather_Conditions
                                          0                                         0
                 Road_Surface_Conditions                  Special_Conditions_at_Site
                                          0                                         0
                     Carriageway_Hazards                          Urban_or_Rural_Area
                                          0                                         0
Did_Police_Officer_Attend_Scene_of_Accident                  LSOA_of_Accident_Location
                                          0                                    129471
```

We find that the following variables included NA values:

- Location_Easting_OSGR
- Location_Northing_OSGR
- Longitude
- Latitude
- Time
- LSOA_of_Accident_Location

We then remove the NA items and duplicated rows in dataset:

```
> df_noNA <- na.omit(df)
> dim(df_noNA)
[1] 1651142      32
> df_clean <- unique(df_noNA)
> dim(df_clean)
[1] 1651142      32
```

Then we needed to establish which variables were necessary to keep in the data frame, we decide to keep the variables which are necessary for mapping and labelling the accidents. Thus, we were left with:

```
> str(Accidents)
Classes 'tbl_df', 'tbl' and 'data.frame':       265078 obs. of  12 variables:
 $ Longitude                 : num  -0.191 -0.205 -0.175 -0.216 -0.213 ...
 $ Latitude                  : num  51.5 51.5 51.5 51.5 51.5 ...
 $ Accident_Severity         : int  2 2 2 2 2 2 2 2 2 ...
 $ Number_of_Casualties      : int  1 1 1 2 1 1 1 1 1 ...
 $ Date                      : chr  "04/01/2005" "20/01/2005" "08/01/2005" "01/02/2005" ...
 $ Day_of_Week               : int  3 5 7 3 3 3 7 3 4 4 ...
 $ Time                      :Classes 'hms', 'difftime'  atomic [1:265078] 63720 900 10800 63000 65700 ...
  .. ..- attr(*, "units")= chr "secs"
 $ Local_Authority_(District): int  12 12 12 12 12 12 12 12 12 12 ...
 $ Light_Conditions          : int  1 4 4 4 4 4 4 1 1 4 ...
 $ Weather_Conditions        : int  2 1 1 2 1 2 2 1 1 1 ...
 $ Road_Surface_Conditions   : int  2 1 1 2 1 2 2 1 1 1 ...
 $ Urban_or_Rural_Area       : int  1 1 1 1 1 1 1 1 1 1 ...
```

We also realised that the dataset could be greatly streamlined if we removed slight accidents, this would also result in our maps consisting of data concerning the most dangerous accidents. Also, to

further shrink the dataset, we removed years before 2010, leaving us with data from 2010-2015. In order for us to produce labels for the data we also needed to include character variables rather than the levels we currently have. So, we merged our data with specific sheets from a guide dataset:

```
Accidents <- left_join(Accidents, AS, by = c("Accident_Severity"))
Accidents <- left_join(Accidents, LA, by = c("Local_Authority_(District)"))
Accidents <- left_join(Accidents, WC, by = c("Weather_Conditions"))
```

In order to be able to add a Year filter to the interactive map, a new variable "Year" was created using the date Variable, the time variable was also adjusted to 12-hour format so that it is more user friendly.

```
library(lubridate)
Accidents$Year = year(as.Date(Accidents$Date, format = "%d/%m/%Y"))
Accidents$Time = format(strptime(Accidents$Time, format='%H:%M:%S'), '%I:%M %p')
```

A label variable was then created for the interactive cluster map. It includes information we feel the user may benefit from, we didn't want to include too much since the label will become unappealing so we chose the variables we felt were most important.

```
library(htmltools)
Accidents$label <- paste("<p>", Accidents$Date, "", Accidents$Time, "</p>",
                         "<p>", Accidents$Severity, "</p>",
                         "<p>", Accidents$Weather, "</p>")
```

Another label was also created for the choropleth using the same method. It included the name of the local authority and the number of casualties for that authority, label was called "label.y".

4.3 Choropleth Map

For the choropleth we wanted to create map showing the total number of casualties for each region. To achieve this, we used the following packages:

```
#load the libraries we need
library(readr)
library(sp)
library(dplyr)
library(rgdal)
library(leaflet)
library(htmltools)
library(mapview)
library(htmlwidgets)
```

We then loaded a shapefile containing the boundaries for each local authority district in Britain (Geoportal.statistics.gov.uk, 2015).

We then use leaflet to draw out the choropleth map:

```
> labels_1 <- paste("<p>", GBLA$lad15nm, "</p>", sep="")
> m  <- leaflet() %>%
+   addProviderTiles(providers$OpenStreetMap) %>%
+   setView(lng = -4.29, lat = 54.09, zoom = 6) %>%
+   addPolygons(data = GBLA,
+               weight = 2,
+               smoothFactor = 0.5,
+               color = "blue",
+               label = lapply(labels_1, HTML))
> m
```

The map is not quite done yet, we must align the variables between the shape file and the dataset:

```
> uk <- Together %>%
+   group_by(label.y) %>%
+   summarise(Num.Casualties = sum(Number_of_Casualties))
> uk1 <- uk[order(match(uk$label.y, GBLA$lad15nm)),]
> n <- is.element(GBLA$lad15nm, uk$label.y)
> summary(n)
   Mode   FALSE    TRUE
logical       3     377
> uk1$label.y
```

Three of the labels do not match the shapefile. We find that label "St Edmunds bury" for GBLA just has different name with uk1's label, so we can change the name of it. The uk1 has no labels "London Airport (Heathrow)" and "Western Isles", so we just remove them:

```
> uk1$label.y[379] <- "St Edmundsbury"
> uk2 <- uk1[order(match(uk1$label.y, GBLA$lad15nm)),]
> n1 <- is.element(GBLA$lad15nm, uk2$label.y)
> summary(n1)
   Mode   FALSE    TRUE
logical       2     378
> uk3 = uk2[c(-379,-380),]
> GBLA1 = subset(GBLA, n1 == TRUE)
> uk4 <- uk3[order(match(uk3$label.y, GBLA1$lad15nm)),]
> n2 <- is.element(GBLA1$lad15nm, uk4$label.y)
> summary(n2)
   Mode    TRUE
logical     378
```

Then we summarise the number of casualties in uk4 data frame and set the colour bins for our leaflet map:

```
> summary(uk4$Num.Casualties)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   39.0   306.0   440.5   506.4   596.8  3216.0
> bins <- c(seq(0,600,100),800,1000,2000,3000, Inf)
> pal <- colorBin("RdYlBu", domain = uk4$Num.Casualties, bins = bins)
```

In the above analysis results, we see that the minimum and maximum number of casualties are 39 and 3216, but the median and mean are 440.5 and 506.4, the 3rd quantile is only 596.8, this means that most of the data is distributed in the range of less than 1000. Based on this data distribution feature, we set colour bins as such:

```
> bins
 [1]    0  100  200  300  400  500  600  800 1000 2000 3000  Inf
```

Finally, we load the data containing the total number of casualties upon the choropleth map, we then add labels and legend to it:

```
> labels_2 <- paste("<p>", uk4$label.y, "</p>",
+                   "<p>", "Number of Casualties: ", round(uk4$Num.Casualties, digits = 3), "</p>",
+                   sep="")
> m  <- leaflet() %>%
+   addProviderTiles(providers$OpenStreetMap) %>%
+   setView(lng = -4.29, lat = 54.09, zoom = 6) %>%
+   addPolygons(data = GBLA1,
+               fillColor = pal(uk4$Num.Casualties),
+               weight = 1,
+               smoothFactor = 0.5,
+               color = "white",
+               fillOpacity = 0.8,
+               highlight = highlightOptions(
+               weight = 5,
+               color = "#666",
+               dashArray = "",
+               fillOpacity = 0.7,
+               bringToFront = TRUE),
+               label = lapply(labels_2, HTML)) %>%
+   addLegend(pal = pal,
+             values = uk4$Num.Casualties,
+             opacity = 0.7,
+             title = NULL,
+             position = "topright")
> m
```

Resulting in the following map:

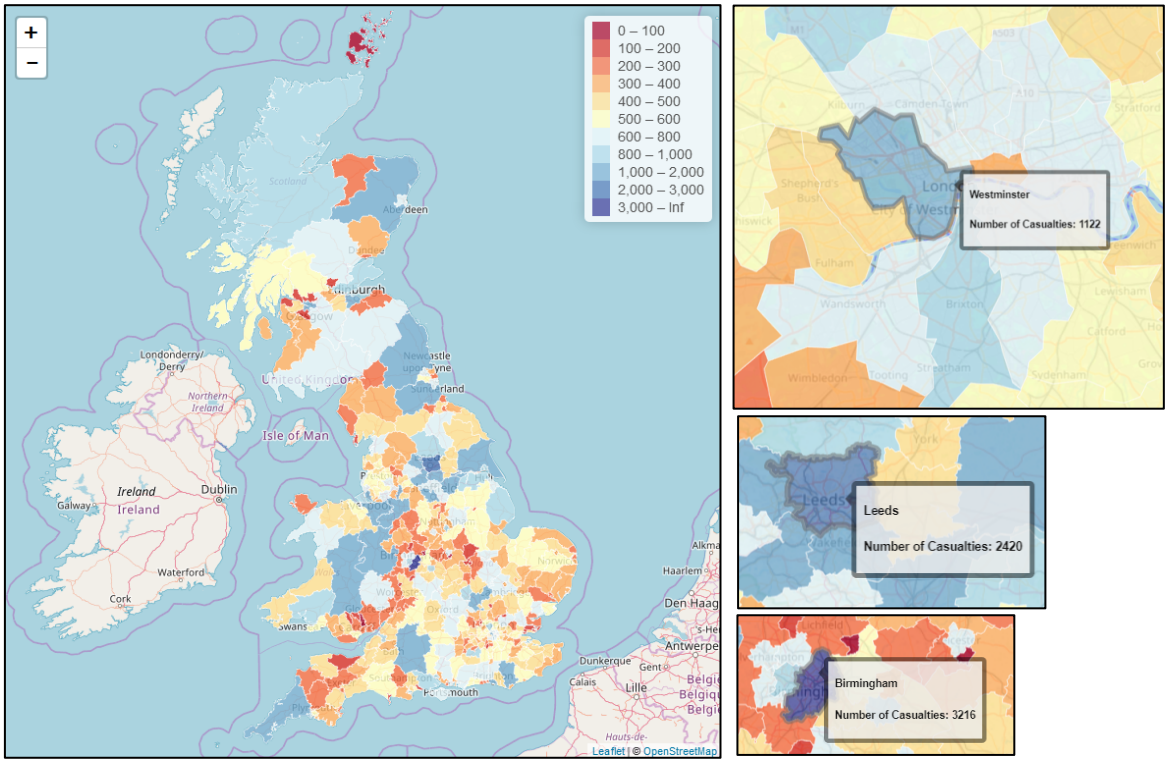

Figure 9

Based on the results shown in the above map, we can summarize that: 1. In most areas of the UK, especially in the East and South, the number of casualties between 2010 and 2015 did not exceed 600 (light yellow areas). 2. In some of the bustling areas, especially the big cities, such as Birmingham, Leeds and Edinburgh, the casualties between 2010-2015 were very large, all over 1,000. Birmingham's casualties are even more than 3,000, which is the highest in all regions. 3. In the northern areas of the UK, the casualties of Highland and Aberdeenshire are the most numerous in the northern regions, it may be related to the topography of these 2 regions and both of them are famous tourist areas. 4. In Central London, casualties in most areas are concentrated in the 600-800 range, and casualties in the surrounding area are also concentrated in the 500-600 range. So, roughly, the sum of casualties in the central London and its suburb areas is highest.

We then add filter year and create a dashboard to present results in different years:

```
# Build another subset including year
uk_1 <- Together %>%
  group_by(Year, label.y) %>%
  summarise(Num.Casualties = sum(Number_of_Casualties))
uk_2 <- uk_1[order(match(uk_1$label.y, GBLA$lad15nm)),]
n_1 <- is.element(GBLA$lad15nm, uk_2$label.y)
summary(n_1)
uk_2$label.y
GBLA$lad15nm
uk_2$label.y[c(2269:2274)] <- "St Edmundsbury"
uk_3 <- uk_2[order(match(uk_2$label.y, GBLA$lad15nm)),]
n_2 <- is.element(GBLA$lad15nm, uk_3$label.y)
summary(n_2)
uk_4 = uk_3[c(-(2269:2280)),]
GBLA_1 = subset(GBLA, n_2 == TRUE)
uk_5 <- uk_4[order(match(uk_4$label.y, GBLA_1$lad15nm)),]
n_3 <- is.element(GBLA_1$lad15nm, uk_5$label.y)
summary(n_3)
uk_5$Year[1409]<-2014
uk_5$Year[1410]<-2015
```

```
# Create a dashboard

bins <- c(seq(0,600,100),800,1000,2000,3000, Inf)
pal <- colorBin("RdYlBu", domain = c(0,1), bins = bins)

UK <- dashboardPage(
  skin = "red",
  dashboardHeader(title = "Casualties Dashboard"),
  dashboardSidebar(
    sliderInput("Year", "Date Range:",
                min = min(uk_5$Year),
                max = max(uk_5$Year),
                value = c(min(uk_5$Year),max(uk_5$Year)),
                sep = "",
                step = 1
    )
  ),
  dashboardBody(
    fluidRow( box(width = 12, length = 100,
                  leafletOutput("mymap")
    )),
    fluidRow( box(width = 12,
                  dataTableOutput("summary_table")))
  )
)
```

```
server <- function(input, output) {
  data_input <- reactive({
    uk_5 %>%
      filter(Year >= input$Year[1]) %>%
      filter(Year <= input$Year[2]) %>%
      group_by(label.y) %>%
      summarise(Num.Casualties = sum(Num.Casualties))
  })

  data_input_ordered <- reactive({
    data_input()[order(match(data_input()$label.y, GBLA_1$lad15nm)),]
  })

  labels <- reactive({
    paste("<p>", data_input_ordered()$label.y, "</p>",
          "<p>", "Number of Casualties: ", round(data_input_ordered()$Num.Casualties, digits = 3), "</p>",
          sep="")})

  output$mymap <- renderLeaflet({
  leaflet() %>%
    setView(lng = -4.29, lat = 54.09, zoom = 6) %>%
    addProviderTiles(providers$OpenStreetMap) %>%
    addPolygons(data = GBLA_1,
                fillColor = pal(data_input_ordered()$Num.Casualties),
                weight = 1,
                smoothFactor = 0.5,
                color = "white",
                fillOpacity = 0.8,
                highlight = highlightOptions(
                  weight = 5,
                  color = "#666",
                  dashArray = "",
                  fillOpacity = 0.7,
                  bringToFront = TRUE),
                label = lapply(labels(), HTML)) %>%
    addLegend(pal = pal,
              values = data_input_ordered()$Num.Casualties,
              opacity = 0.7,
              title = NULL,
              position = "topright")
  })

  output$summary_table = DT::renderDataTable(
    data_input(), options = list(lengthChange = FALSE)
  )

}

shinyApp(ui = UK, server = server)
```

Finally, we run the app code and show the results for different years:

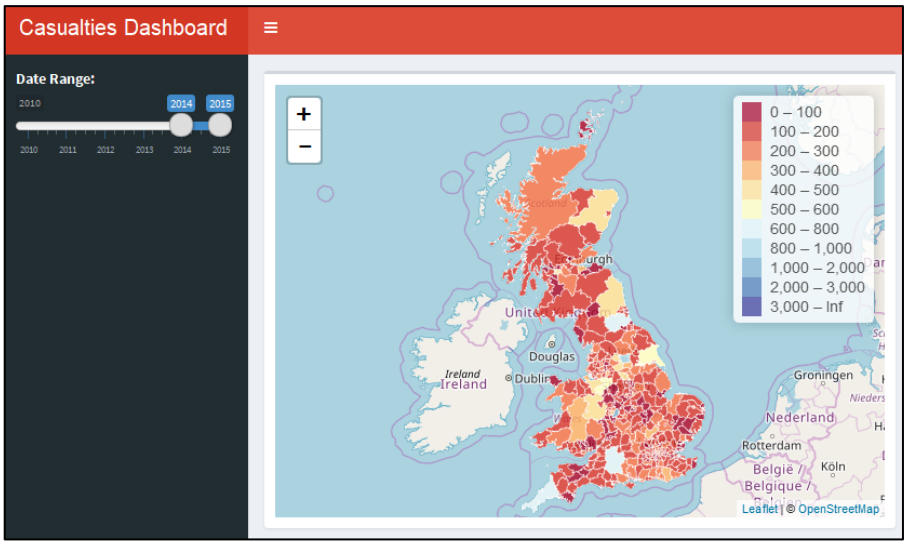

Figure 10

4.4 Interactive Cluster Map

The purpose of creating this map is to produce a user-friendly tool capable of displaying the exact location of each fatal and severe accident. We used leaflet to create the map and then shiny to present it in a user-friendly manner. The same packages were used for this as the choropleth.

We first began by creating a leaflet map and adding markers for the longitude and latitude, these markers were then clustered. A search bar and labels were also added.

```
m <- leaflet() %>%
  addTiles() %>%
  addMarkers(lng = Accidents$Longitude, lat = Accidents$Latitude,
             clusterOptions = markerClusterOptions(showCoverageOnHover = FALSE),
             label = lapply(Accidents$label, HTML)) %>%
  addSearchOSM()
```

Then we created the shiny app, we also added a year slider:

```
Accidents <- read_csv("~/Desktop/LDP/Accidents.csv")
ui <- dashboardPage(
  skin = "red",
  dashboardHeader(title = "Traffic Accidents"),
  dashboardSidebar(
    sliderInput("date_range", "Year",
                min = min(Accidents$Year),
                max = max(Accidents$Year),
                value = c(min(Accidents$Year),max(Accidents$Year)),
                sep = "",
                step = 1
    )
  ),
  dashboardBody(
    fluidRow( box(width = 12,
                  leafletOutput("mymap")
    ))
  )
)

server <- function(input, output) {

  data_input <- reactive({
    Accidents %>%
      filter(Year >= input$date_range[1]) %>%
      filter(Year <= input$date_range[2])
  })
  output$mymap <- renderLeaflet({
    data <- data_input()
    leaflet() %>%
      addTiles() %>%
      addMarkers(lng = data$Longitude, lat = data$Latitude,
                 clusterOptions = markerClusterOptions(showCoverageOnHover = FALSE),
                 label = lapply(data$label, HTML)) %>%
      addSearchOSM()
  })


}

shinyApp(ui, server)
```

This resulted in the following map shown in three views:

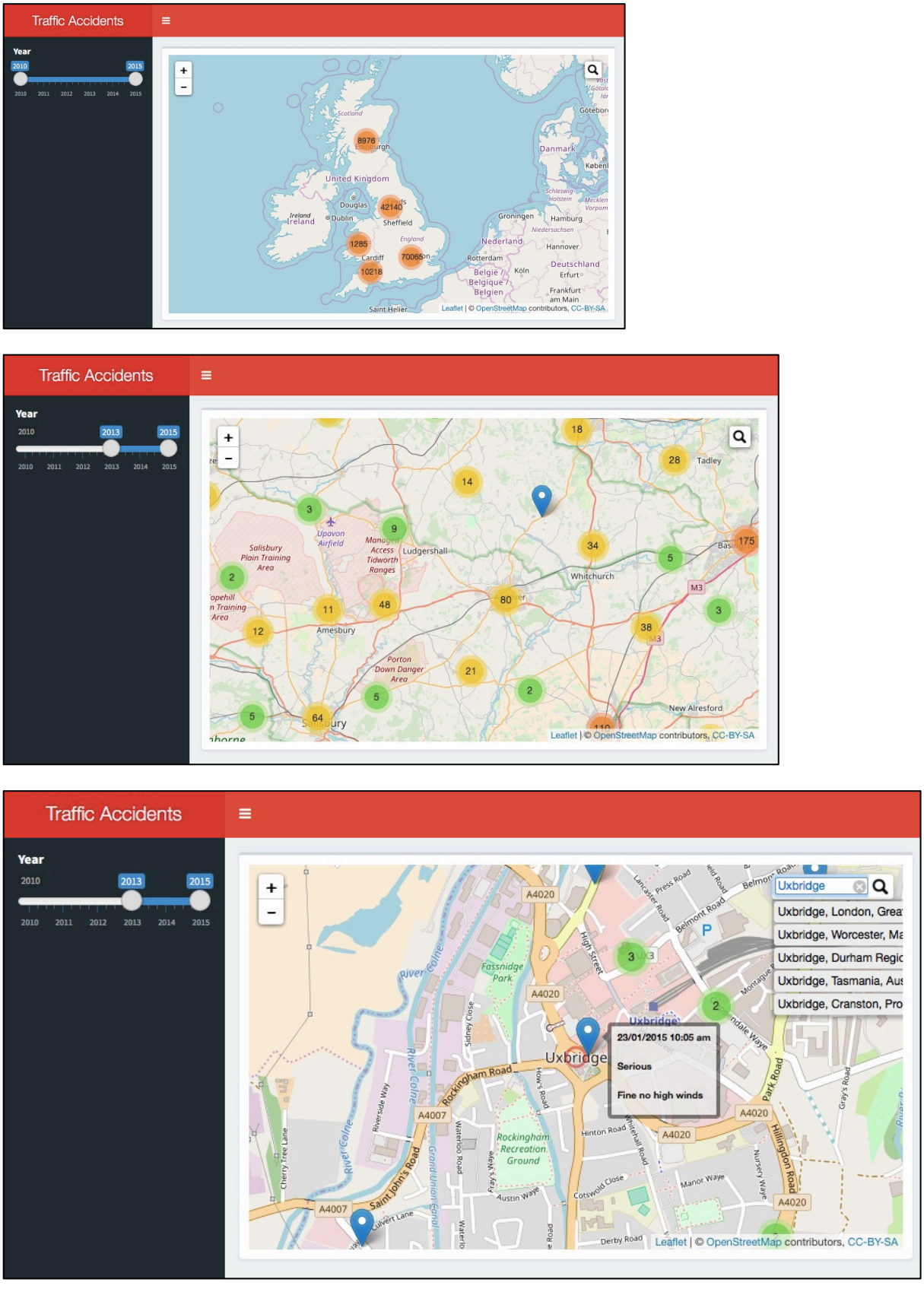

Figure 11

The visualisation uses clustering to reduce the number of markers seen in one frame. Clicking on a cluster will zoom in to it, creating additional smaller clusters and markers which are colour coded depending on the size of the cluster. The search bar has an autocomplete function, which is handy for users, it also highlights the selected area. Hovering over the selected marker previews the label.

The year can be changed using the slider on the left panel, the user can select a single year or a range of years.

Chapter 5 Machine Learning Predictions

5.1 Introduction

The purpose of this chapter is to develop a model that helps predict accident severity (*accident severity is defined by three values: 1 = Fatal Accident, 2 = Serious Accident and 3 = Slight Accident*) and the number of casualties. The reason for this is to discover the conditions which lead to dangerous accidents, thereby, helping potential travellers make wiser decisions in the future. We use principle component analysis, Random Forest, Decision Tree model, Multiple linear regression, K-nearest neighbours (KNN), Neural Network and Logistic Regression.

5.2 Principle Component Analysis (PCA)

We begin by conducting a principal component analysis to identify correlations between the independent variables. To do this we created scatter charts:

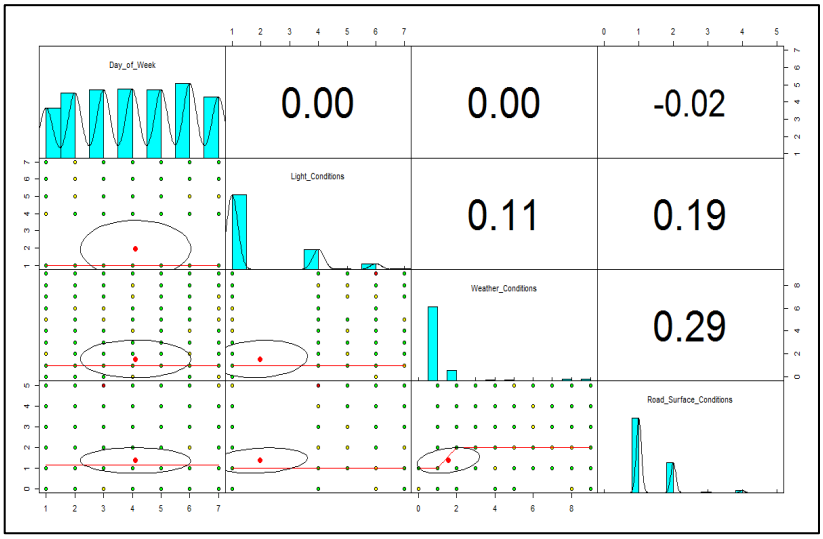

Figure 12

The scatter charts show the correlation relationship between independent variables. A positive correlation between light& weather conditions. Road surface conditions have a correlation closet to negative zero. However, Day of week correlation seems to be low. When developing a predictive model and the correlation among independent variables are high, this cause "multicollinearity "problem. Not all charts give a positive 0 or negative 0, such as the correlation of road surface conditions, Although the correlation closed to negative 0, however, it also has a value of 0.19 and 0.29, these values are a bit far from zero when compared to other variables in the graph plots. Estimate for the model are unstable, hence the prediction will not be accurate. Principal component analysis helps to handle this problem. Principal component is done only on the independent variables, before principal component is done, all the variables need to be normalized.

PC is then printed:

```
Standard deviations:
[1] 1.1847298 1.0003852 0.9471796 0.8357605

Rotation:
                             PC1          PC2          PC3          PC4
Day_of_Week            -0.02623809  0.996129475 -0.07371405 -0.04004835
Light_Conditions        0.47330207  0.084503232  0.84366409  0.23890430
Weather_Conditions      0.59524143  0.002897662 -0.51005084  0.62090852
Road_Surface_Conditions 0.64883308 -0.024018239 -0.15048352 -0.74551557


> sd(training$Day_of_Week)
[1] 1.923962

> summary(pc)
Importance of components:
                          PC1    PC2    PC3    PC4
Standard deviation     1.1847 1.0004 0.9472 0.8358
Proportion of Variance 0.3509 0.2502 0.2243 0.1746
Cumulative Proportion  0.3509 0.6011 0.8254 1.0000
Importance of components:
```

By printing pc, information such as standard deviation, rotation also referred to as loadings. because four variables were stored in principal component, four pcs are printed which represent the four variables. Each of this pcs are a normalized linear combination of original variables listed below. Rotation or loading are the coefficient of the linear combination of the continuous variables. Each of these values below, will lie between positive and negative 1. This value indicate that principal component increases so as the Light conditions, weather conditions and road surface conditions also increased, meanwhile as pc1 increases, day of week decreases due to a negative correlation. The highest value is present in road surface conditions (0.64883308), while the lowest value in terms of correlation is day of week with a value of (-0.02623809). in pc2, the highest value is positive value of (0.996129475), and the lowest value in terms of correlation is road surface conditions (-0.024018239). this means that when pc2 increases, the day of week, light conditions, weather conditions all increased, that is positive correlation, but as pc2 increases, road surface conditions decreased, negative correlation. As pc3 increases so is light conditions, while day of week, weather conditions and road surface conditions decrease giving negative correlations, however, light conditions increases as pc3 increases giving a positive correlation between pc3 and light conditions.

As pc4 increases, light conditions and weather conditions also increases as pc4 increases, while day of week and road surface conditions decreases as pc4 increases. In pc4, road surface conditions have a value of -0.74551557 which is closed to negative 1, hence road surface conditions are mainly characterized by road surface conditions.

Looking at the summary of the principle component, PC4 explain about 35% of the proportion of variance variability, pc2 captured 25% of the variability, pc3 captured 22% of the variability while pc4 only 17% of the variability. The variability continues to go down.

For the cumulative proportion, pc2 already gave about 60% of the variability is been explained, pc3 and pc4 do not play any important role in terms of variability. However, pc1 and pc2 did captured majority of the variability.

5.3 Random Forest Analysis

The random forest method is developed by aggregating trees. Random Forest, instead of creating one decision tree, hundreds of decisions can be created. Then by aggregating the results from all the trees, a classification model is produced. Random forest can used for both classification and regression. If Y variable is a factor variable, classification model will be applied, however, if Y

variable is continuous variable then regression be applied. Random forest has the advantage of not overfitting, it also supports a large number of variables. Random forest also aids with variable selections depending on the importance of each one of the variables, this helps with the overall accuracy of the model.

For the random forest we partition the data into training and test sets in the ratio of 70:30, respectively. We then set Accident Severity as the dependent variable. Day of Week, Light Conditions, Number of Records, Road Surface Conditions and Weather Conditions are the independent variables. We use the program R to carry out the random forest model on the training set:

```
Set.seed(333)
> rf <- randomForest(train$Accident.Severity~., data=train)
> print(rf)

Call:
 random Forest (formula = train$Accident.Severity ~ ., data = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of error rate: 14.78%
Confusion matrix:
  1 2    3  class.error
1 0 0  102 1.0000000000
2 0 2  938 0.9978723404
3 0 2 6004 0.0003330003
```

Information about the developed model is generated above, the used data is from the training data and the random forest used is a classification model because accident severity is a factor variable. The default number of trees is 500, and the number of variables tried at each split is 2. Ideally the default is around square root of p, where p is the number of variables, because the number of variables is 6, the closest value is 2. The Out of bag estimate error rate is 14.78% (about 15% misclassification error), which is good. About 85% accuracy, which is a decent result. The confusion matrix, a very good prediction when predicting class 1, however, the prediction is very high when predicting class 2, error rate in class 3 is a bit high.

We create a confusion matrix for the training set:

```
> confusionMatrix (p1, train$Accident.Severity)
Confusion Matrix and Statistics

               Accuracy: 0.8529   Accuracy is quite high, which is good
                 95% CI :(0.8444, 0.8611) confidence interval quite tight.
    No Information Rate: 0.8522
    P-Value [Acc > NIR] : 0.4415

                  Kappa : 0.0082
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1  Class: 2
Sensitivity           0.00000 0.0053191
Specificity           1.00000 1.0000000
Pos Pred Value            NaN 1.0000000
Neg Pred Value        0.98553 0.8672441
Prevalence            0.01447 0.1333712
Detection Rate        0.00000 0.0007094
Detection Prevalence 0.00000 0.0007094
Balanced Accuracy     0.50000 0.5026596
```

Sensitivity is not good for predicting class 1, considering the above sensitivity value of 0.0000000. class 2 sensitivity a bit poor, however class 3 sensitivity gives a good value.

```
                     Class: 3
Sensitivity          1.000000
Specificity          0.004798
Pos Pred Value       0.852762
Neg Pred Value       1.000000
Prevalence           0.852157
Detection Rate       0.852157
Detection Prevalence 0.999291
Balanced Accuracy    0.502399
```

Sensitivity for class 1 is 0.0000, this calculate how often class 1 was correctly classified as 1 or predicted as 1. In this case, correct classification in class 1 is 0.0000, this means that class 1 was not correctly classified, similarly for class 2, 0.005, this is the number of times class 2 was correctly classified finally, was classified 1:0000, this is the number of times class 3 was correctly classified.

```
Confusion Matrix
                 Reference
Prediction     1    2    3
         1     0    0    0
         2     0    5    0
         3   102  935 6006
```

The model prediction class 1 corresponded with the reference (Actual classes for the response variable). So, we have Accident Severity level 1, 2, 3 and the predicted values are 1, 2, 3. 6006 times experts classified the severity of accident as 3 and the model also predict it to be 3, so this is a correct classification. Similarly, there is 5 correct classification for class 2 and 0 classification for class 1.

We then predict the model, on the test set:

```
# Prediction & Confusion Matrix - Test data
> p2 <- predict(rf, test)
> confusionMatrix(p2, test$Accident.Severity)

Confusion Matrix and Statistics

          Reference
Prediction    1    2    3
         1    0    0    0
         2    0    0    1
         3   35  411 2505

Overall Statistics

               Accuracy : 0.8486
                 95% CI : (0.8351, 0.8613)
    No Information Rate : 0.8489
    P-Value [Acc > NIR] : 0.5331

                  Kappa : -6e-04
 Mcnemar's Test P-Value : NA

> rf$confusion
  1 2    3  class.error
1 0 0  102 1.0000000000
2 0 2  938 0.9978723404
3 0 2 6004 0.0003330003
```

The test prediction two for class 3 was correctly classified 2505 times compared to class 1 and class 2, these two classes were misclassified, giving them a value of 0 classification. once again accuracy rate of 85% Is good, it means that the model accurately classified the some of the classes correctly, although one of the class was misclassified.

We then plot the error rate:

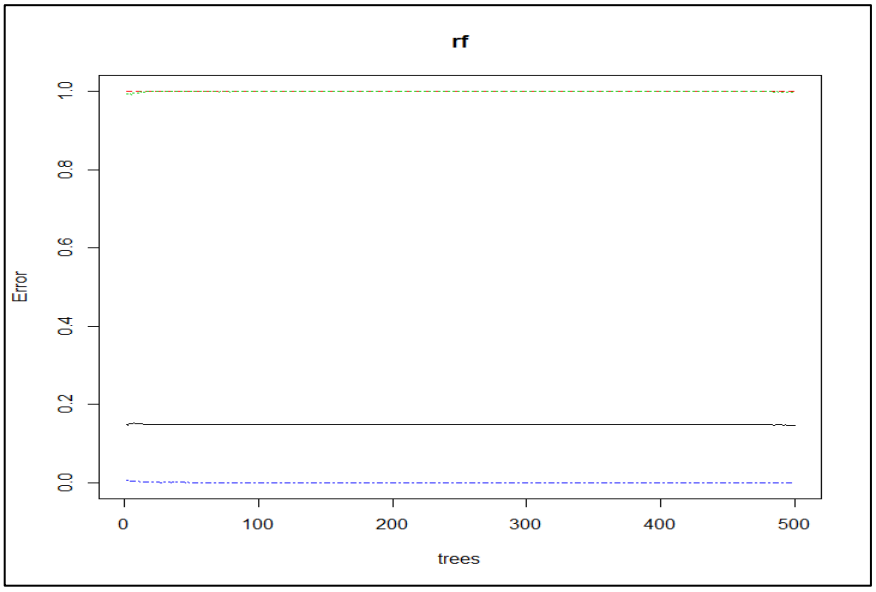

Figure 13

Figure 13: The plot seems to indicate that after 100 decision trees, there is not a significant reduction in error rate. Unpruned tree is built by choosing best split, based on a random sample of mtry predictors at each node. new data using majority votes for classification prediction.

We then tune the random forest model and plot it again:

```
>  t <- tuneRF(train[,-6], train[,6], stepFactor = 1, plot = TRUE, ntreeTr
y = 200, trace = TRUE, improve = 0.5 )
>  Plot(rf)
```

We then plot a histogram for the number of nodes for the trees:

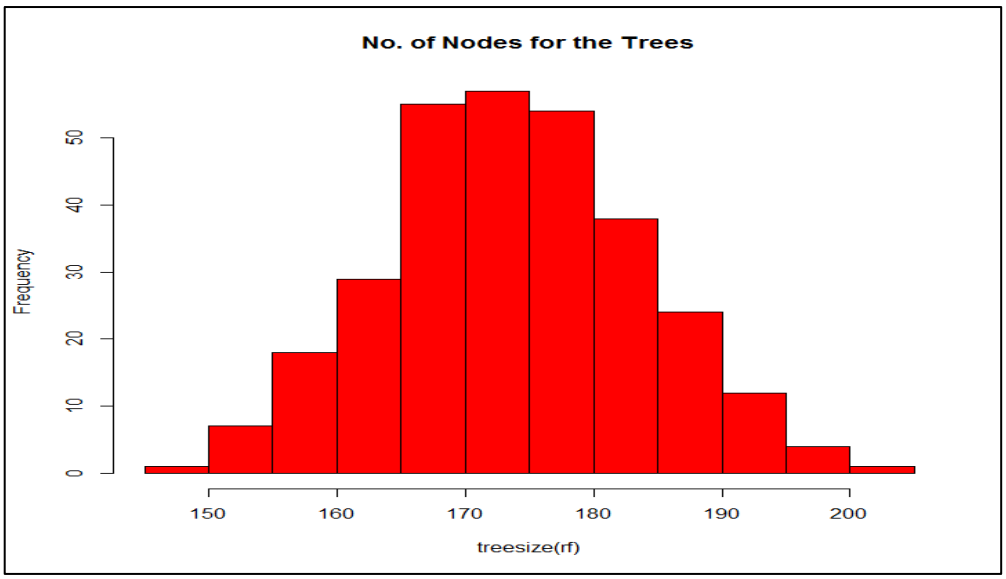

Figure 14

The histogram plot gives the size of tree in terms of number of nodes. The histogram gives the distribution of the number of nodes in each of the 300 trees that was generated. There are 3 high peaks on the y-axis of the histogram that are above 50, meaning that they have more than 50 trees that contained nodes in them. Also, on the left side of x-axis, there are over 150 trees that contained

nodes in them and in the right side of the x-axis, there are over 200 trees that contained nodes in them. Overall, distributions of nodes in these 300 trees are from about 150 to 200. The vast majority of the trees have close to over 50 nodes.

We now want to calculate the importance of the variables in the model:

```
>  rf <- randomForest(train$Accident.Severity~.,data = train, ntree = 300,
mtry = 4, importance = TRUE, proximity = TRUE)
>  varImpPlot(rf, main = "Importance. of Variables in the Model", col = "r
ed")
```

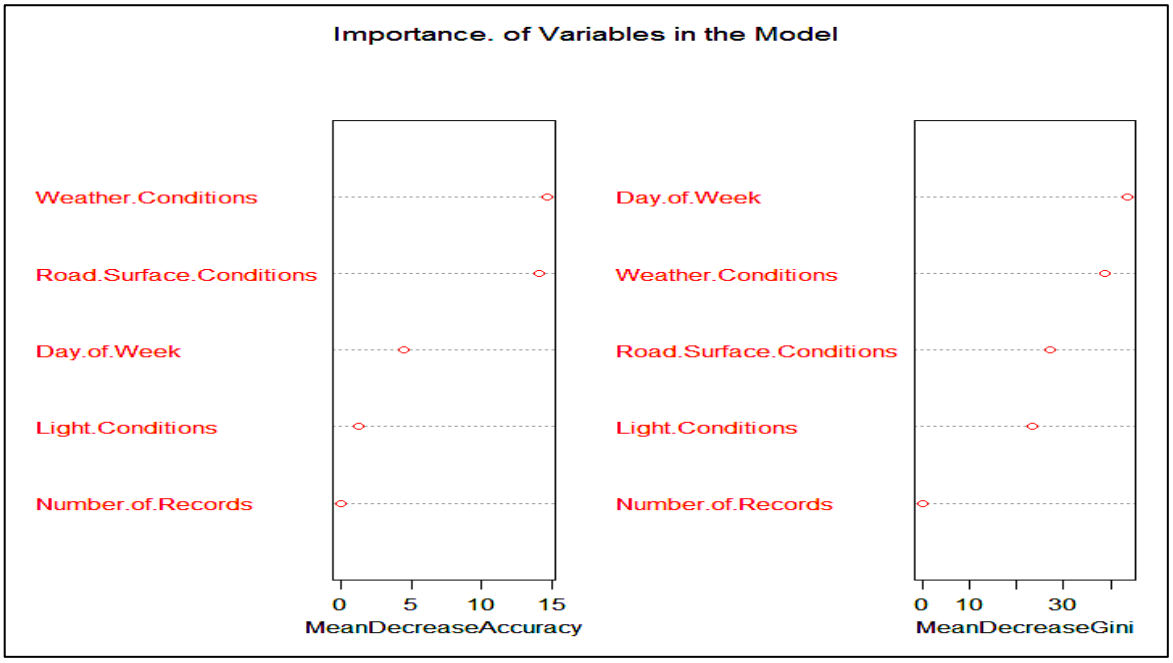

Figure 15

Variable importance, shows which variables play an important role in the model. Chart 1 (MeanDecreaseAccuracy); This chart tests how much worse the model performs without each variable. Indicating that if weather conditions is removed whilst building the tree, the mean will decrease in accuracy. Because weather condition has a very high value that means it has maximum importance in terms of contributing to accuracy. The next variable importance is the road surface conditions, also is day of week high variable importance. Compared the three variables, number of record is almost to 0, hence number of record variable shows less importance for prediction. Chart 2 (MeanDecreaseGini), This chart measures how pure the nodes are at the end of the tree without each variable. For example, when one of the Important variable Is removed, how much on an average gini decreases. Day of week has the highest contribution towards this parameter. when removed this high contributing variable, It will affect the perfomance of this analysis, Therefore, day of week, weather conditions, road surface conditions and light conditions stand out of the 5 variables.

Overall, the model prediction and accuracy was not too far from very good, 85% accuracy Is good, although the sensitivity rate In both class 1 & 2 were very poor when compared to sensitivity rate for class 3. The model achieved high accuracy rate and good prediction values, also It's ability to successfully to classified the vairiables to Its target.

5.3 Decision Tree

A decision tree is both a classifier and predicting model, this strategic decision support tool is used to identify possible outcome of an event occurring or not. This model is built based on the whole dataset, by utilizing the entire variables of interest. The decision tree model was used alongside other machine learning tools to identify variables that may contribute to the response variable (That is the target variable). In order words, variables that plays important role towards the response variable.

Assigning the target or response variable as a factor:

```
Training.dataa$Accident.SeverityF<-factor (Training.dataa$Accident.Severity)
```

A new variable is created within the data file, named accident. severity (F for factor), changed integral variable into a factor or categorical variable by using the above code. Once ran the code, the observation numbers increased from 5 to 6 variables.

Data was partitioned into Training and Validation datasets:

```
pd <- sample(2, nrow(Training.dataa), replace = TRUE, prob = c(0.8,0.2))
```

- ```
  tree <- ctree(Training.dataa$Accident.SeverityF~Training.dataa$Day.of.Week+Train
  ing.dataa$Light.Conditions+Training.dataa$Number.of.Records+Training.dataa$Road.
  Surface.Conditions+Training.dataa$Weather.Conditions, data = train)
  ```
- ```
  tree
  ```

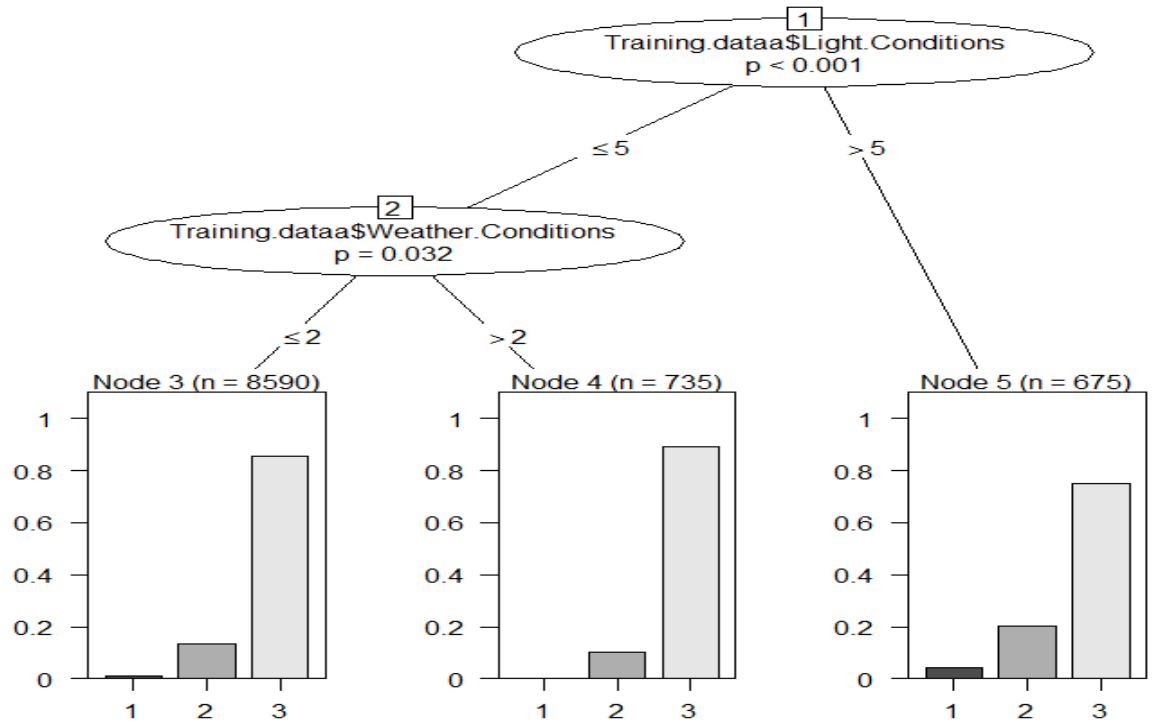

Figure 16

The most important variable to the prediction model is right at the top, which is light conditions. Light conditions are the most important variables out of the five variables in classifying the observations into 1,2, and 3. where 1 from the response variable represent fatal, 2 represent serious and 3 represent slight. If light conditions are < or = 5, then branch to the left side of the tree otherwise branch right, similarly, if weather conditions is < or = 2, then go to terminal node. The terminal node is the decision nodes which looks at the probability where response variable = 1, indicate that the light conditions have high influence on accident fatality in terminal node 3. In this case, the probability that both light and weather conditions contributed to accident fatality is between 0 to 5% chance, meanwhile between 0 to 15% chance for it getting to serious accident and over 80% chance of been slight. In terminal node 4, if weather conditions are greater than 2, the response level 2 has the highest probability that weather conditions would be a contributing factor to slight occurrence of accident severity, very close to close to probability of 1. Although these variables may not have significant influence on accident severity, however, they do have some influence on the response variable that is accident severity. The decision tree model did somewhat agree with the variable importance prediction of random forest prediction, although it was only able to identify only two variables importance in its prediction unlike random forest which was able to identify 4 important variables and indicate the variables that contributed less in the prediction. Light conditions contributed about 97% in the decision tree predictions.

From the validated test output, three results are produced [1] 0.01147453, [2] 0.13018767, and [3] 0.85833780. From the three level of response variable, where 1 represent fatal, 2 serious, and 3 slight.

1. indicate the probability that both light conditions and weather conditions influenced the response variable that is accident severity at a probability rate of 0.01147453.

2. indicate the probability that both light and weather conditions influenced the response variable at a probability rate of 0.13018767.

3. Indicate the probability that both light and weather conditions influenced the response variable slightly at a probability rate of 0. 85833780.The probability that the two variables influenced the response variable slightly is very high, although these variables may not be the main contributing factors, never the less, there is about 86% chance of a slight contribution of these variables towards the cause of accident severity.

5.5 Multiple Linear Regression

The advantage of using a regression model is that it's an easy way to analyse a multivariate model. A regression model can clearly show the fitting relationship between the dependent variable and independent variables. This model is also very straightforward to interpret. For this model, we have accident severity as the dependent variable

The formula used is shown below:

Accident severity = β0longitude + β1latitude + β2day of week +β3area + β4light condition + β5weather condition + β6road surface condition + β7urban or rural area + ϵ

```
Call:
lm(formula = accident_clean$accident.Accident_Severity ~ accident_clean$accident.Longitude +
    accident_clean$accident.Latitude + accident_clean$accident.Day_of_Week +
    accident_clean$accident.Local_Authority_.District. + accident_clean$accident.Light_Conditions +
    accident_clean$accident.Weather_Conditions + accident_clean$accident.Road_Surface_Conditions +
    accident_clean$accident.Urban_or_Rural_Area)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9661  0.1166  0.1361  0.1838  0.3989

Coefficients:
                                                Estimate Std. Error  t value Pr(>|t|)
(Intercept)                                    3.359e+00  1.127e-02  297.952  < 2e-16 ***
accident_clean$accident.Longitude             -1.983e-03  2.539e-04   -7.811 5.67e-15 ***
accident_clean$accident.Latitude              -8.185e-03  2.162e-04  -37.858  < 2e-16 ***
accident_clean$accident.Day_of_Week            4.599e-04  1.555e-04    2.957  0.00311 **
accident_clean$accident.Local_Authority_.District. -2.833e-05  1.298e-06  -21.818  < 2e-16 ***
accident_clean$accident.Light_Conditions      -1.580e-02  1.855e-04  -85.159  < 2e-16 ***
accident_clean$accident.Weather_Conditions     6.157e-03  1.901e-04   32.387  < 2e-16 ***
accident_clean$accident.Road_Surface_Conditions  1.820e-02  5.108e-04   35.629  < 2e-16 ***
accident_clean$accident.Urban_or_Rural_Area   -6.552e-02  6.511e-04 -100.644  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3992 on 1780644 degrees of freedom
Multiple R-squared:  0.01299,   Adjusted R-squared:  0.01299
F-statistic:  2930 on 8 and 1780644 DF,  p-value: < 2.2e-16
```

From the result, we obtain the multiple linear regression equations:

Accident severity = (-1.983e-03)longitude + (-8.185e-03)latitude + (4.599e-04)day of week +(-2.833e-05)area + (-1.580e-02)light condition + (6.157e-03)weather condition + (1.820e-02)road surface condition + (-6.552e-02)urban or rural area + 3.359e+00

Most variables are significant, which are in less than 0.1% significant level. Only one variable day of week is 0.01% significant level. Next, we drop the variable day of week and try to perform a new linear regression.

```
Call:
lm(formula = accident_clean$accident.Accident_Severity ~ accident_clean$accident.Longitude +
    accident_clean$accident.Latitude + accident_clean$accident.Local_Authority_.District. +
    accident_clean$accident.Light_Conditions + accident_clean$accident.Weather_Conditions +
    accident_clean$accident.Road_Surface_Conditions + accident_clean$accident.Urban_or_Rural_Area)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9671  0.1170  0.1362  0.1839  0.4002

Coefficients:
                                                Estimate Std. Error  t value Pr(>|t|)
(Intercept)                                    3.361e+00  1.126e-02  298.560  < 2e-16 ***
accident_clean$accident.Longitude             -1.984e-03  2.539e-04   -7.816 5.47e-15 ***
accident_clean$accident.Latitude              -8.183e-03  2.162e-04  -37.848  < 2e-16 ***
accident_clean$accident.Local_Authority_.District. -2.831e-05  1.298e-06  -21.807  < 2e-16 ***
accident_clean$accident.Light_Conditions      -1.579e-02  1.855e-04  -85.131  < 2e-16 ***
accident_clean$accident.Weather_Conditions     6.157e-03  1.901e-04   32.389  < 2e-16 ***
accident_clean$accident.Road_Surface_Conditions  1.818e-02  5.108e-04   35.598  < 2e-16 ***
accident_clean$accident.Urban_or_Rural_Area   -6.556e-02  6.510e-04 -100.708  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3992 on 1780645 degrees of freedom
Multiple R-squared:  0.01299,   Adjusted R-squared:  0.01299
F-statistic:  3348 on 7 and 1780645 DF,  p-value: < 2.2e-16
```

Then I run a code to determine which model is better:

```
anova(Severity_model,Severity_model1)

Analysis of Variance Table

Model 1: accident_clean$accident.Accident_Severity ~ accident_clean$accident.Longitude +
    accident_clean$accident.Latitude + accident_clean$accident.Day_of_Week +
    accident_clean$accident.Local_Authority_.District. + accident_clean$accident.Light_Conditions +
    accident_clean$accident.Weather_Conditions + accident_clean$accident.Road_Surface_Conditions +
    accident_clean$accident.Urban_or_Rural_Area
Model 2: accident_clean$accident.Accident_Severity ~ accident_clean$accident.Longitude +
    accident_clean$accident.Latitude + accident_clean$accident.Local_Authority_.District. +
    accident_clean$accident.Light_Conditions + accident_clean$accident.Weather_Conditions +
    accident_clean$accident.Road_Surface_Conditions + accident_clean$accident.Urban_or_Rural_Area
   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1 1780644 283731
2 1780645 283733 -1    -1.393 8.7425 0.003109 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the p-value is 0.003109, which shows significant evidence that the new model is better than the old one. The multiple linear regression equations are shown below:

Accident severity = (-1.984e-03)longitude + (-8.183e-03)latitude + (-2.831e-05)area + (-1.579e-02)light condition + (6.157e-03)weather condition + (1.818e-02)road surface condition + (-6.556e-02)urban or rural area + 3.361e+00

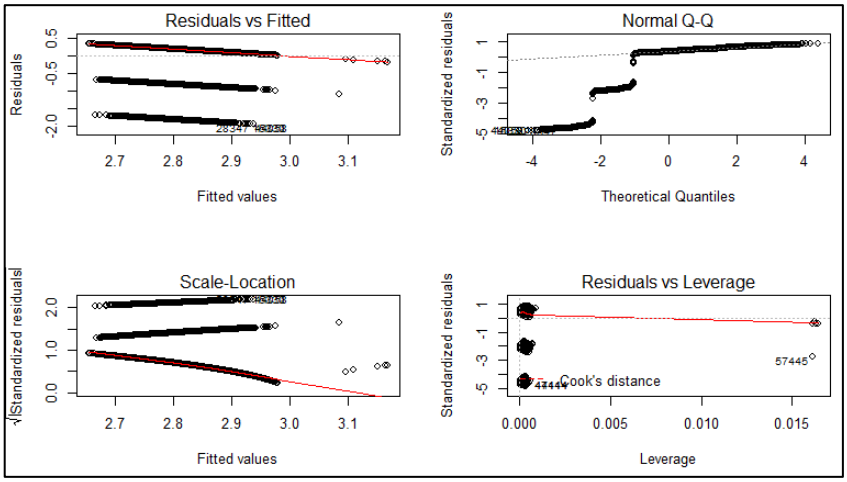

Figure 17

The top left figure shows the point concentrate fitted value in 2.7-3.0. The Q-Q plot shows most of residual from point distributed on the line. The bottom left and bottom right figures show that the data is concentrated in the three-main area. The large number of outline point distribution cause the weakness of linear model. The estimate value we calculate from linear model could have a big deviation.

Then i use this model to predict test dataset and correlate the result and predicted result.

```
prob_pred = predict(Severity_model, newdata = test_set[,-(3:4)])
prob_pred1 = predict(Severity_model1, newdata = test_set[,-(3:4)])
```

```
> cor(predit_results$actual, predit_results$pred)
[1] 0.113994
> cor(predit_results$actual, predit_results$pred1)
[1] 0.1139727
```

We can find the accuracy is really low, it is only about 11%. Hence, the multiple linear regression model is not reliable.

5.6 KNN

Then, I try to use another machine learning method--KNN to do prediction for variable accident severity. The advantage of KNN method is that the KNN method is a good way to predict big dataset. It is sample and fast to training. It is good for both classification and regression model.

```
MinMax <- function(x){
  tx <- (x - min(x)) / (max(x) - min(x))
  return(tx)
}
accident_clean_minmax <- apply(accident_clean, 2, MinMax)
accident_clean_minmax <- as.data.frame(accident_clean_minmax)
```

First, I create the minmax function and use it to calculate the dataset. It could make dataset much easier to do calculation.

```
k_value = sqrt(dim(training_accident_clean_minmax)[1])
```

Next set the k-value, and do the prediction

```
        knn
actual      0   0.5     1
    0       0     0   342
    0.5     0     0  3732
    1       0     0 22636
```

The result is shown below:

```
> acc_knn <- sum(diag(knn_results_table)) / sum(knn_results_table)
> acc_knn
[1] 0.8474729
```

We can know the predict result is confirm to the actual in accuracy. The result shows the accuracy is 84.74%. the KNN model is a sustainable model to predict accident severity.

5.7 Neural Networks

In order to create a prediction model for "number of casualties" as the dependent variable we use the method of Neural Networks. This is because it has a strong distribution processing capacity and it also has a strong ability to learn.

We begin by conducting a linear regression model to better understand the relationship between the dependent variable and the independent variables:

```
Call:
lm(formula = accident_clean$accident.Number_of_Casualties ~ accident_clean$accident.Longitude +
    accident_clean$accident.Latitude + accident_clean$accident.Day_of_Week +
    accident_clean$accident.Local_Authority_.District. + accident_clean$accident.Light_Conditions +
    accident_clean$accident.Weather_Conditions + accident_clean$accident.Road_Surface_Conditions +
    accident_clean$accident.Urban_or_Rural_Area)

Residuals:
      Min        1Q    Median        3Q       Max
-0.778148 -0.375806 -0.275888 -0.201855 91.533873

Coefficients:
                                                 Estimate      Std. Error   t value             Pr(>|t|)
(Intercept)                                   0.663484491782  0.023075636943  28.75260 <0.0000000000000002 ***
accident_clean$accident.Longitude            -0.019899966317  0.000519632444 -38.29624 <0.0000000000000002 ***
accident_clean$accident.Latitude              0.007305616594  0.000442524090  16.50897 <0.0000000000000002 ***
accident_clean$accident.Day_of_Week           0.000327716808  0.000318365080   1.02937               0.3033
accident_clean$accident.Local_Authority_.District. -0.000101467131  0.000002657332 -38.18383 <0.0000000000000002 ***
accident_clean$accident.Light_Conditions      0.013967290266  0.000379666482  36.78832 <0.0000000000000002 ***
accident_clean$accident.Weather_Conditions   -0.007214259395  0.000389109194 -18.54045 <0.0000000000000002 ***
accident_clean$accident.Road_Surface_Conditions  0.014121686329  0.001045542193  13.50657 <0.0000000000000002 ***
accident_clean$accident.Urban_or_Rural_Area    0.201230207871  0.001332614611 151.00405 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8170545 on 1780644 degrees of freedom
Multiple R-squared:  0.01618956,        Adjusted R-squared:  0.01618514
F-statistic:  3662.78 on 8 and 1780644 DF,  p-value: < 0.00000000000000022204
```

We can see all the variables have very strong relationship with number of casualties except day of week. Thus, I try to build two different Neural Network model, one has variable day of week, one has not.

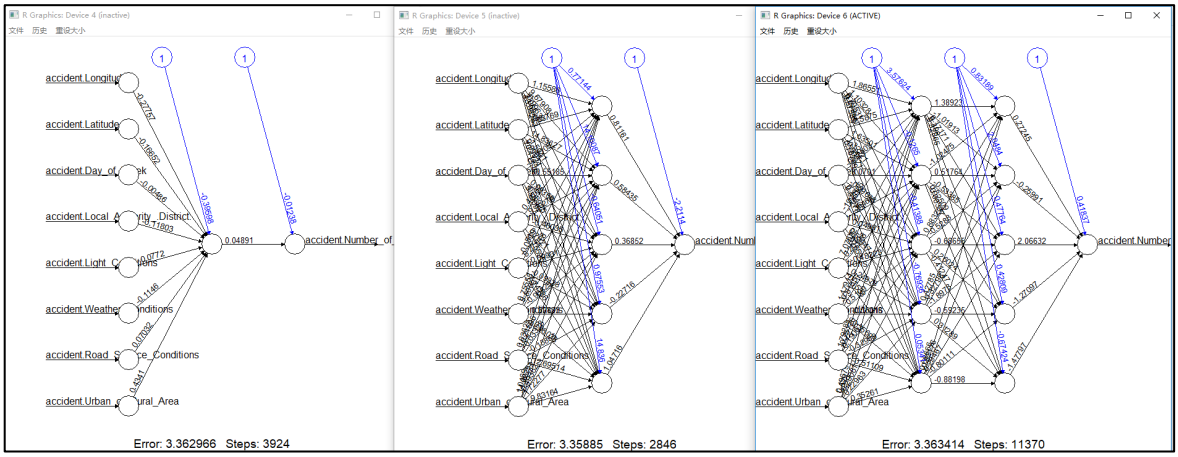

Figure 18

This figure is of three Neural Network models, which are 1 hidden node with variable day of week, 5 hidden nodes on 1 layer with variable day of week and 5 hidden nodes on 2 layers with variable day of week.

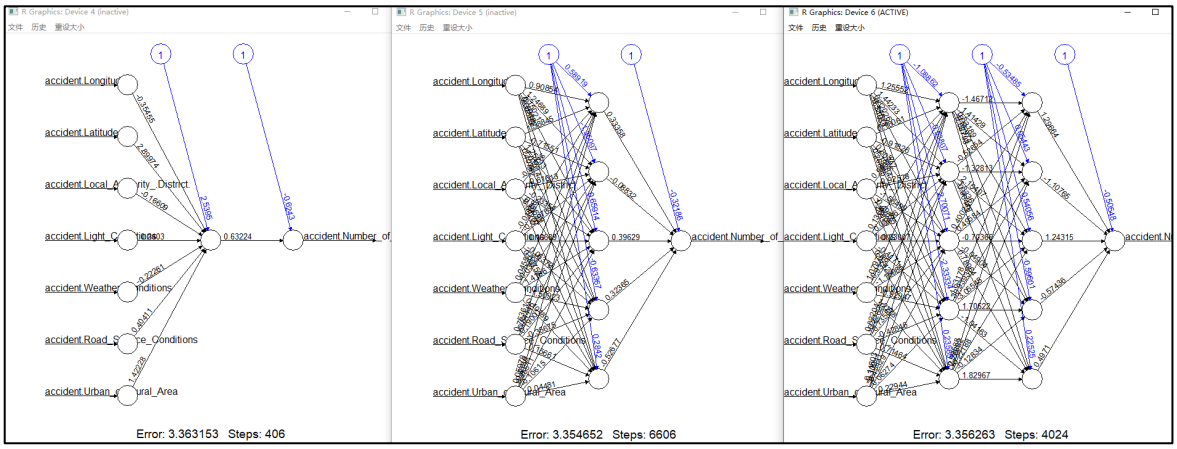

Figure 19

This figure is of three neural network models without variable day of week. The first one is 1 hidden node. The second is 5 nodes on 1 layer. The third is 5 nodes on 2 layers.

```
> pred_accident_Casualties_nn_1 <- compute(accident_Casualties_nn_1, test_accident_clean_minmax[,-(3:4)])
> pred_accident_Casualties_nn_5 <- compute(accident_Casualties_nn_5, test_accident_clean_minmax[,-(3:4)])
> pred_accident_Casualties_nn_55 <- compute(accident_Casualties_nn_55, test_accident_clean_minmax[,-(3:4)])
> pred_accident_Casualties_nn_11 <- compute(accident_Casualties_nn_11, test_accident_clean_minmax[,-(3:5)])
> pred_accident_Casualties_nn_51 <- compute(accident_Casualties_nn_51, test_accident_clean_minmax[,-(3:5)])
> pred_accident_Casualties_nn_551 <- compute(accident_Casualties_nn_551, test_accident_clean_minmax[,-(3:5)])
> accident_Casualties_results <- data.frame(
+   actual = test_accident_clean_minmax$accident.Number_of_Casualties,
+   nn_1 = pred_accident_Casualties_nn_1$net.result,
+   nn_5 = pred_accident_Casualties_nn_5$net.result,
+   nn_55 = pred_accident_Casualties_nn_55$net.result,
+   nn_11 = pred_accident_Casualties_nn_11$net.result,
+   nn_51 = pred_accident_Casualties_nn_51$net.result,
+   nn_551 = pred_accident_Casualties_nn_551$net.result
+ )
> cor(accident_Casualties_results[,'actual'], accident_Casualties_results[,c("nn_1","nn_5", "nn_55","nn_11","nn_51", "nn_551")])
           nn_1         nn_5        nn_55        nn_11        nn_51        nn_551
[1,] 0.123109724 0.1262122144 0.121013621 0.122960428 0.1337544403 0.1298424171
```

Then, we use these six models to predict the test dataset and compare with the actual result. We can see that all the models perform poor in prediction. All the model predicted result only have 12%-14% correlation with actual result. The highest correlation is 5 nodes on 2 layers without variable day of week Neural Network model.

We can know the predict result have large range deviation. The small value has good prediction. The possible reason that caused the model weakness is the sample size limitation. The dataset is big

enough, but the personal computer can't do full data in neural network. Hence, I just use some of data from dataset to build this model. that's the main reason for sample size limitation.

5.8 Logistic Regression

Since many of the categorical variables are represented in levels, it is difficult to identify the specific conditions that cause dangerous accidents. Thus, we adapt the dataset accordingly and perform a logistic regression.

We begin by preparing the data, first we expand the variable "Accident Severity" in excel by creating an additional variable labeled "Serious_and_Fatal". We then use a binary method to allocate 1 for Serious and Fatal accidents and 0 for Slight accidents. This excel sheet is then merged with our dataset in R using the left_join function. This will be our dependent variable since it will help us determine the conditions that result most dangerous accidents.

Accident_Severity	Severity	Serious_and_Fatal
1	Fatal	1
2	Serious	1
3	Slight	0

Table 2

The same method was used to expand Weather_Conditions, Light_Conditions and Road_Surface_Conditions, resulting in the binary variables: Rain, Snow_or_Fog, Light, Dry_Road, Snow_Icy_Road and Flooded_Road. Below is the summary of the Logistic Regression model:

```
Call:
glm(formula = Serious_and_Fatal ~ ., family = binomial, data = training_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2461  -0.5380  -0.5345  -0.5221   2.2165

Coefficients:
                    Estimate Std. Error  z value Pr(>|z|)
(Intercept)         -1.600274   0.015140 -105.701  < 2e-16 ***
Number_of_Casualties  0.241260   0.003434   70.251  < 2e-16 ***
Day_of_Week         -0.003478   0.001657   -2.099   0.0358 *
Rain                -0.166174   0.012203  -13.617  < 2e-16 ***
Snow_or_Fog         -0.161375   0.030448   -5.300 1.16e-07 ***
Light               -0.540621   0.011026  -49.034  < 2e-16 ***
Dry_Road             0.043256   0.008970    4.823 1.42e-06 ***
Snow_Icy_Road       -0.276593   0.022766  -12.149  < 2e-16 ***
Flooded_Road         0.159152   0.082063    1.939   0.0525 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 652790  on 786430  degrees of freedom
Residual deviance: 645089  on 786422  degrees of freedom
AIC: 645107

Number of Fisher Scoring iterations: 4
```

From the summary we can draw some conclusions about the data. All the variables except for Day_of_week and Flooded_Road are significant for predicting fatal and serious accidents due to their P values.

We can see that rain has a negative relationship with the dependent variable and since rain is equal to 1, we can say that accidents that occur during the rain are more likely to be slight. Snow_or_Fog also has a negative relationship and since snow is equal to 1 and fog is equal to 0, this indicated that an accident in the fog is more likely to result in a dangerous accident.

Light has a strong negative coefficient and since light is equal to 1 and darkness is equal to 0, so this indicates that if there is accident when there's light outside a slight accident is more likely, whereas if there is darkness a dangerous accident is likely.

Dry_road has a positive coefficient, however, since the value is incredibly small this doesn't indicate much about dangerous accidents.

Snow_Icy_Road, has a negative estimate, and since snow is equal to 1 and Icy is equal to 0, we can say that accidents in snowy roads are likely to result in slight accidents and accidents in Icy roads are likely to be dangerous accidents.

The final variable is Flooded_Road and it has a positive coefficient, and since flooded road is equal to 1, we can conclude that accidents in flooded roads are likely to be dangerous accidents.

5.9 Machine Learning Evaluation

Of all the machine learning methods we used, KNN and Random Forest were the most successful for predicting Accident Severity, with accuracy of 84.74% and 85% respectively. The multiple linear regression was the least successful with an accuracy of just 11%. The Logistic regression along with the random forest helped answer the question of the most dangerous conditions sufficiently, with the random forest indicating that weather conditions and road surface conditions being the most important variables when predicting accident severity, and the logistic regression revealing the specific conditions which lead to serious and fatal accidents.

However, because to the large size of the original data, an immense processing power would be required in order to run the model. Therefore, with all the models a subset of the data was used. This may have reduced the accuracy of the models.

Chapter 6 Conclusion

The main goals for this study were to find the most dangerous areas and conditions for traveling by road in the UK. The reason being to help people avoid dangerous traffic accidents in the future. We aimed to answer these questions through exploratory analysis, spatial analysis and machine learning predictions. We also aimed to provide users with a tool to help them visualise accident locations in the UK.

Through the exploratory analysis, we found that the conditions which were most likely to result in dangerous traffic accidents are:

- Light Conditions: Darkness
- Day of the Week: Saturday and Sunday
- Road Surface Conditions: Flooded Roads, Snowy Roads and Icy Roads
- Weather: Snowing and Foggy or Misty

With the spatial analysis we were able to find the areas that resulted in the most accidents through a choropleth, we found that big urban cities such Birmingham, Leeds, Edinburgh and London had the largest number of casualties. We were also able to create an interactive cluster map intended provide insight for those seeking a more specific overview of the location of past accidents, this will aid people who are nervous about travelling, it will also help when moving home.

Through the machine learning models, we found that random forest and KNN are the most accurate predictors of traffic accident severity. We found out that weather conditions and road surface conditions are most important in predicting traffic accidents. Through the Logistic regression, we found that darkness, icy roads and foggy/misty weather are the most influential determinants of serious and fatal accidents.

In the future, we would like to expand on our project by creating an application capable of accurately predicting the likelihood of a traffic accident through user inputted conditions and location. We would also like to create a routing application that calculates the safest route rather than the fastest route.

References

- Abdalla, I., Raeside, R., Barker, D. and McGuigan, D. (1997). An investigation into the relationships between area social characteristics and road accident casualties. Accident Analysis & Prevention, 29(5), pp.583-593.
- Abdelwahab, H. and Abdel-Aty, M. (2001). Development of Artificial Neural Network Models to Predict Driver Injury Severity in Traffic Accidents at Signalized Intersections. Transportation Research Record: Journal of the Transportation Research Board, 1746, pp.6-13.
- Geoportal.statistics.gov.uk. (2015). [online] Available at: http://geoportal.statistics.gov.uk/datasets/local-authority-districts-december-2015-full-clipped-boundaries-in-great-britain.
- Kaggle.com. (2018). UK Car Accidents 2005-2015 | Kaggle. [online] Available at: https://www.kaggle.com/silicon99/dft-accident-data [Accessed 13 Apr. 2018].
- Liu, C.H., Sharma, A. 2017, "Exploring spatio-temporal effects in traffic crash trend analysis", Analytic Methods in Accident Research, vol. 16, no. 12, pp. 104-116.
- Ma, X.X., Chen, S.R., Chen, F. 2017, "Multivariate space-time modeling of crash frequencies by injury severity levels", Analytic Methods in Accident Research, vol.15, no. 9, pp. 29-40.
- Mannering, F. 2018, "Temporal instability and the analysis of highway accident data", Analytic Methods in Accident Research, vol.17, no. 3, pp. 1-13.
- RAC (2018). Road safety questions and answers. [online] Racfoundation.org. Available at: https://www.racfoundation.org/motoring-faqs/safety#a1 [Accessed 13 Apr. 2018].
- Richard, R. and Ray, S. (2017). A tale of two cities: Analyzing road accidents with big spatial data. 2017 IEEE International Conference on Big Data (Big Data).

App links:

https://ukaccidentscasualtiesmaps12.shinyapps.io/Special_Analysis/

https://ukaccidentscasualtiesmaps12.shinyapps.io/Special_Analysis_Choropleth/