Exploring the use of decision trees and random forest in producing road traffic crash severity models

Road traffic accidents are currently one of the biggest concerns for global health organisations and national governments, particularly concerning the level of injury caused. Understanding the factors which contribute to accident severity and using these to predict accident severity for new crashes is hugely researched. Classification techniques are a popular choice for researching this area. This case study investigated the ability of decision trees and random forest models to classify accident severity using UK traffic accident data from 2016 and 2017. Random forest models were found to perform better than decision trees. Common variables found to be linked to accident severity were lighting, speed limit, junction detail and urban or rural environment. Although the models did not have a high enough accuracy to be used to classify accidents in real-time, the variables identified by the models can be used to inform further investigation.

1. Introduction

The increasing numbers of motor vehicles across the globe has become a major concern for health organisations and national governments. In 2018, the global number of annual road traffic deaths had reached 1.35 million, with road traffic-related injuries being the leading cause of death for people aged between 5 and 29 years (World Health Organization, 2018). The World Health Organisation has declared that drastic action is required and in 2020, they set a target of halving road traffic deaths by 2030 as part of their resolution A/RES/74/299 "Improving global road safety". They aim to achieve this through improving road and vehicle design, laws and immediate emergency care for injured parties. Utilising crash severity prediction models, where accurate models can be used to predict the event of serious injury and/or fatality in a crash based on known characteristics, could enhance efforts to meet this goal. Knowing what crash attributes are important in determining crash severity outcome can allow more targeted safety designs and law changes. Additionally, a live model able to predict injury severity based on these reported attributes could notify hospitals more quickly of potential incoming traumas, allowing more immediate treatment of casualties.

Road traffic crash severity models have been studied widely for a number of years. Initially statistical models were used to classify accident severity and determine important contributors to accident severity. Some of the most utilised methods included multinomial logit, ordered logit, ordered probit, binary logit, binary probit and nested logit (reviewed in detail by Savolainen et al. (2011)). However, statistical models are limited due to pre-defined underlying relationships between the dependent variable (accident severity) and the independent variables (potential accident risks) and model assumptions; for example, assuming independent variables are independent of one another (Chang and Wang, 2006). This has led to a rise in the popularity of applying machine learning methods, as these non-parametric models do not assume any relationship between variables or distribution (Wen et al., 2021). Classification is a form of supervised machine learning. A training dataset, where each sample has already been assigned a class, is used to build a model that can map predictor attributes to the class attributes appropriately. A testing dataset with known class attributes can then be used to test the accuracy of the resulting model’s ability to assign class attributes. Many classification methods have been used to classify crash severity, including decision trees, random forest, Bayes classifier, support vector machine and artificial neural networks (Silva et al., 2020).

Decision trees (DT) can be a useful tool for looking at factors which contribute to the severity outcome of a road accident and how these factors interact with each other. Plots of decision trees allow users to visually trace a pathway through to each factor that produces a particular severity outcome and so are more easily interpretable than some other methods (Silva et al., 2020). Decision trees have been utilised by many researchers for crash severity modelling because of the interpretability. For example, Pakgohar et al. (2011) looked at road accident data in Iran and identified driving licence and safety belts as having an important role in accident severity. When comparing the results to logistic regression, they found the decision tree’s results more accurate and easier to interpret. However, decision trees can be prone to overfitting. Random forests (RF) can overcome this limitation with its robustness against overfitting and lack of sensitivity to any outliers (Breiman, 2001). This technique aggregates the modelling results of multiple DT models so often has better predictability. When comparing RF models to CART models in their ability to predict injury severity in traffic accidents, Krishnaveni and Hemalatha (2011) found that the RF model had better accuracy. Similarly, RF outperformed the C4.5 model when predicting injury severity for different driver groups (Mafi et al., 2018).

The aim of this study is to measure the ability of DT and RF models to produce a crash severity classification model, using measures such as model accuracy. The output and accuracy of these models will also be compared to one another to investigate any differences in the two methods. The data used is from the UK Department for Road Transport and includes details of UK road accidents reported by the police in the years 2016 and 2017. Two pieces of software which can be used for developing machine learning models are R and SAS enterprise miner. R is a free open-source software for statistical computing, with a variety of add-on packages which provide many built-in functions with a variety of add-on packages which provide many built-in functions, including datamining techniques (R Core Team, 2021). SAS enterprise miner is a software which claims to be able to “streamline the data mining process to develop models quickly” (SAS Institute Inc., 2015). This tool aims to allow non-technical users with limited statistical skill to easily generate models and results which are easy to interpret. An additional aim of this study is to compare the results generated using DT and RF by these two platforms.

2. Literature review

When investigating and modelling traffic accident severity using data mining, a number of different methods have been applied due to their ability to outperform statistical methods. Li et al. (2012) compared the performance of support vector machine (SVM) models against an orbit probit model in predicting injury severity. The SVM model produced better prediction results and more reasonable results for lower frequency injury classes. SVM modelling was also applied to accident severity prediction for a mountainous freeway section. Real-time traffic and weather data were incorporated into the model, which were found to have large influence on crash severity (Yu and Abdel-Aty, 2014). Another comparative study also found that SVM models performed well, along with Random Forest, when compared to Mutinomial Logit and Nearest Neighbour Classification (Iranitalab and Khattak, 2017). Applications of Bayesian Networks identified that the type of accident, the age of the driver and lighting were associated with fatal and serious injury accidents in Spain (De Oña et al., 2011). Application of association rules mining is very useful for uncovering hidden relationships between independent factors and traffic accident severity. For severe traffic accidents in China, complex interactions between driver behaviour, vehicle type, road characteristics and environmental factors were identified (Xu et al., 2018). Using UK data, Feng et al. (2019) suggested that road type, light, speed limit and road surface are influential factors for traffic accidents after applying association rules mining. Applying ANN to crashes among elderly drivers identified the cause of collision, average annual daily traffic, number of vehicles, age, road surface condition and gender as important contributory factors (Amiri et al., 2020). Zeng and Huang (2014) identified sex, age, wearing a seatbelt or not, vehicle age, point of impact and number of heavy vehicles as influential on crash severity in Florida.

One of the earliest data mining methods implemented was using a decision tree. Chang and Wang (2006) used CART to establish any relationships between accident severity and various independent variables. They identified vehicle type as being the most important variable associated with crash severity, with motorcycle riders being more likely to be seriously injured in an accident. Another study using CART found that lack of a seatbelt, improper overtaking and speeding affected injury severity in Iran (Tavakoli Kashani et al., 2011). Supporting these findings, another study identified that human factors had the highest contribution to traffic accidents (Pakgohar et al., 2011). Using other decision tree methods has also produced similar results, with seatbelt usage, collision type and use of drugs influencing injury severity in car accidents using a C5 model (Delen et al., 2017). Many of the studies which use RF to classify accident severity are comparing the results of different models. Krishnaveni and Hemalatha (2011) found that the RF model had better accuracy compared to CART models in their ability to predict injury severity in traffic accidents. RF also outperformed the C4.5 model when predicting injury severity for different driver groups (Mafi et al., 2018). RF was found to have the highest accuracy (53.9%) compared to SVM, k-nearest neighbours and DT (Zhang et al., 2018) and again had a higher accuracy than SVM in another study but was not as accurate as nearest neighbour classification (Iranitalab and Khattak, 2017). When using RF to predict motorcycle crash severity, (Wahab and Jiang, 2019) found that the most important factors for influencing severity were location type, time of crash, settlement type, collision partner, collision type, road separation, road surface type, day of week, year and round shoulder condition.

3. Methodology

3.1 Dataset acquisition and preparation

The dataset used in this investigation was obtained from the UK’s Department for Road Transport government website. The dataset includes details of road accidents collected by the police using the STAT19 reporting system between 2016 and 2020 (Department for Public Transport, 2021a). This reporting system is used to standardise the process of collecting details regarding a reported road accident involving at least one vehicle collision or a vehicle collision with a pedestrian. The initial dataset contains 597,973 recorded incidents, each with 36 attributes recorded. Each accident is given an accident severity level of either slight, severe or fatal (Department for Transport, 2021). A slight accident includes at least one person with a slight injury (an injury of minor character, such as bruises, cuts or sprains) but no further injury or death. A serious accident includes at least one person who is seriously injured (an injury requiring hospital in-patient treatment) but no one is killed. A fatal accident includes at least one death and potentially other injuries. The first step prior to modelling requires cleaning and preparation of the dataset into an appropriate format. This process was completed using R version 4.1.2 (R Core Team, 2021), which the following section will describe, and full detail of the process and code used can be found in Appendix A.

Due to the size of the dataset and lack of computing power, only the first two years (2016 and 2017) were used for this analysis, reducing the number of accidents to 266,603. Initial brief exploration of the dataset was first conducted, using functions such as ‘summary’ and ‘head’. Each accident was attributed a unique identifier (renamed to ‘accident\_index’) and initial checks found there was not any duplicates. With the focus being to identify causes of different accident severities, some attributes could be removed due to irrelevance [fig.1]. For example, details such as the likely road number and local highway authority would likely not contribute to an accident’s outcome.

> #remove initially selected unnecessary variables

> #remove accident reference, police force, local\_authority\_district,

> #local\_authority\_ons\_district, local\_authority\_highway, first\_road\_number,

> #second\_road\_number, lsoa\_of\_accident\_location, did\_police\_officer\_attend\_scene\_of\_accident

> accident\_df <- accident\_df %>% select(-c(accident\_reference, police\_force, local\_authority\_district, local\_authority\_ons\_district, local\_authority\_highway, first\_road\_number, second\_road\_number, lsoa\_of\_accident\_location, did\_police\_officer\_attend\_scene\_of\_accident))

+ }

+ }

**Figure 1:** R code used to remove unnecessary variables (accident\_reference, police\_force, local\_authority\_district, local\_authority\_ons\_district, local\_authority\_highway, first\_road\_number, second\_road\_number, lsoa\_of\_accident\_location, did\_police\_officer\_attend\_scene\_of\_accident).

Dealing with missing data is hugely important for data mining and modelling, as randomly missing data can interfere with decision tree and random forest effectiveness. Although it is likely that any missing data is due to lack of information when reporting an accident, it is possible that any missing information could not be random and therefore may be informative. In this dataset, missing data was generally given either the value ‘-1’ or ‘NULL’. These were changed to ‘NA’ to allow simpler detection of missing data. The function ‘vis\_miss’ from the package ‘visdat’ (Tierney, 2017) was used to visualise the missing data [fig.2]. Fortunately, only 3.3% of data was missing altogether. The missing data does not appear to follow a pattern. The columns ‘second\_road\_class’ and ‘junction\_control’ both have around 48% missing values each, leading to the decision to remove these columns.

A picture containing table

Description automatically generated

**Figure 2:** Plot of observations for each variable where black indicates observations where data is missing (NA).

The remaining accident attributes were all in numerical form, with nominal attributes given a number (for example, in accident\_severity 1 was assigned to fatal accidents), allowing potential correlations between variables to be investigated. Any strong correlations could interfere with modelling. However, to reduce dimensionality and simplify analysis, ‘time’ and ‘date’ were first changed into ‘hour’ and ‘month’ respectively. ‘hour’ was produced by removing the minutes from the hour, so ‘15:57’ would become ‘15’. To produce ‘month’, the strings were converted into dates (‘%d/%m/%Y’) and then reformatted to just show the month (‘%m’). Then a correlation matrix was produced using the package ‘corrplot’ (Wei and Simko, 2021) [fig. 3]. There does appear to be some correlation between ‘urban\_or\_rural’ and ‘speed\_limit’ and between ‘trunk\_road\_flag’ and ‘speed\_limit’, however it was decided that this correlation was not strong enough to warrant removal of any of these attributes.

Chart, scatter chart

Description automatically generated

**Figure 3:** Correlation matrix between variables where colour indicates correlation value and size indicates strength of correlation.

The next and final stage involves visualising the distribution of the ‘accident\_severity’ values within each individual attribute. This allows investigation of potentially interesting patterns or necessary altering of attributes to improve analysis. Numerical attributes, including ‘location\_northing\_osgr’, ‘location\_easting\_osgr’, ‘latitude’, ‘longitude’, ‘hour’, ‘month’, ‘number\_of\_vehicles’ and ‘number\_of\_casualties’, were plotted as histograms [fig.4]. Figure 4A and figure 4C show that ‘location\_northing\_osgr’ and ‘latitude’ have an identical distribution and figure 4B and figure 4D show that ‘location\_easting\_osgr’ and ‘longitude’ also have an identical distribution. This makes sense as ‘location\_northing\_osgr’ and ‘location\_easting\_osgr’ are the recorded locations of the accidents according to the Ordnance Survey National Grid reference system used by the UK. Therefore, therefore accidents at the same site will have matching longitude/latitude coordinates along with matching Ordnance Survey National Grid references. As only one of these measurements is required for location, ‘location\_northing\_osgr’ and ‘location\_easting\_osgr’ were removed from the dataset. The number of accidents per each hour is interesting, as there appears to be high numbers of accidents in the morning (approximately between 7:00 and 10:00) and in the evening (approximately between 15:00 and 19:00) [fig.4E]. These times coincide with rush hour where there are often higher levels of traffic on the roads, which would explain the increased likelihood of a car accident. There does not appear to be any large differences in the accident severity or frequency of accidents between the months or any obvious patterns [fig.4F]. The number of casualties [fig.4G] and the number vehicles [fig.4H] involved in accidents are both skewed to the left, with lower values having higher frequencies. There does not appear to be any interesting differences between the accident severities of these values [fig.4G and fig.4H].

Chart, histogram

Description automatically generated

Chart, bar chart, histogram

Description automatically generated

**Figure 4:** Histograms for numeric variables of accidents. **A)** Northing coordinates. **B)** Easting coordinates. **C)** Latitude. **D)** Longitude. **E)** Hour. **F)** Month. **G)** Number of causalities. **F)** Number of vehicles.

As the nominal attributes were provided in numeric format, interpreting the differences between attribute variables in any plots would be more difficult, therefore it is necessary to convert the numbers into a factor level which better implies the corresponding value. For example, the value 7 in ‘weather\_conditions’ was changed to ‘FOG’. A supporting document was provided on the same website to understand each category, including the numbers substituted for categorical data (Department for Public Transport, 2021b). Each attribute received the same treatment; checking for missing values, conversion from numbers to meaningful values, conversion to a factor and then a check to ensure each value has been successfully changed. The column ‘weather\_conditions’ received further altering. Most of the weather conditions were split into ‘condition’ and ‘condition with high winds’, such as ‘rain’ and ‘rain with high winds’. It was decided to create a new column for the presence of high winds and combine all of the same weather conditions together. It could be possible that high winds could have an increasing effect on its own or have different effects according to weather combination. Figure 5 shows the shared and novel treatments for ‘weather\_conditions’ as an example. For details of conversion of the other attributes look in Appendix A.

> #weather\_conditions

> #check NA

> sum(is.na(accident\_df$weather\_conditions))

[1] 14

> #convert to NA

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==9]<-NA

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==8]<-NA

> #check levels

> accident\_df %>% count(weather\_conditions)

weather\_conditions n

1 1 216951

2 2 28792

3 3 987

4 4 2503

5 5 2488

6 6 178

7 7 1369

8 NA 13335

>

> #high\_winds

> accident\_df$high\_winds<-NA

>

> for(i in 1:nrow(accident\_df)) {

+ if (is.na(accident\_df$weather\_conditions[i])) {

+ accident\_df$high\_winds[i] <- NA

+ }

+ else if ((accident\_df$weather\_conditions[i]<4) || (accident\_df$weather\_conditions[i]==7)){

+ accident\_df$high\_winds[i] <- "NO"

+ }

+

+ else if ((accident\_df$weather\_conditions[i]>3) & (accident\_df$weather\_conditions[i]<7)){

+ accident\_df$high\_winds[i] <- "YES"

>

> accident\_df %>% count(high\_winds)

high\_winds n

1 NO 248099

2 YES 5169

3 <NA> 13335

>

> #combine high winds and non high-winds

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==4]<-1

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==5]<-2

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==6]<-3

>

> #change level names

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==1]<-"FINE"

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==2]<-"RAIN"

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==3]<-"SNOW"

> accident\_df$weather\_conditions[accident\_df$weather\_conditions==7]<-"FOG"

> #check levels

> accident\_df %>% count(weather\_conditions)

weather\_conditions n

1 FINE 219454

2 FOG 1369

3 RAIN 31280

4 SNOW 1165

5 <NA> 13335

**Figure 5:** R code for conversion of weather condition variables (‘weather\_conditions’) into more informative values and creating the ‘high\_winds’ variable.

Bar charts were produced to visualise the spread of accident severity classes within each accident attribute [fig.6, fig.7, fig.8]. Rather than use absolute frequencies, relative frequencies were used to allow comparison across characteristics, as some characteristics may be less common than others. Code for the plots was adapted from Julia Silge’s blog (Silge, 2021). Speed limit (‘speed\_limit’) was treated as a factor as, despite being an integer, speed limits for roads are discrete variables. The frequency of the different accident severities did not appear to change between the two years [fig.6A]. The day of the week did not seem to have different levels of accident severity, except maybe a slightly higher proportion of fatal and serious accidents occurring on the weekend [fig.6B]. The junction detail, road type and speed limit have seemingly visible differences when it comes to serious and slight accidents. Vehicles not at a junction seemed have a higher proportion of serious and fatal accidents whereas vehicles on slip roads comparatively had less serious accidents but similar levels of fatal accidents [fig.6D]. Single carriageways and double carriageways have comparatively high levels of fatal and serious accidents [fig.6E]. As speed limit increases, so does the amount of serious and fatal accidents, reaching a peak at 60mph and decreasing slightly at 70mph [fig.6F]. The road class does not appear to have as obvious a relationship with accident severity compared to the previous three mentioned [fig.6C]. Areas without lighting visibly appear to have a much higher proportion of fatal accidents [fig.7A]. Roads without crossing facilities altogether seems to have more fatal accidents [fig.7B and fig.7C]. There does not appear to be a large difference between the accident severity proportions according to the different weather conditions [fig.7D]. Existing accidents acting as a road hazard appear to have a higher proportion of fatalities than other hazards and lack of road sign appears to have a higher proportion of serious accidents [fig.8A and fig.8D]. Additionally, urban areas appear to have a higher proportion of serious and fatal accidents [fig.8F]. However, the presence of high winds and the presence of special conditions at the site did not appear to have different levels of accident severities [fig.B and fig.C].

Graphical user interface, bar chart

Description automatically generated

**F**

**E**

**D**

**C**

**B**

**A**

**Figure 6:** Proportion of accident severities (fatal, serious or slight) for categorical variables. **A)** accident year **B)** day of week **C)** first road class **D)** junction detail **E)** road type **F)** speed limit

Graphical user interface, bar chart

Description automatically generated

**D**

**C**

**B**

**A**

**Figure 7:** Proportion of accident severities (fatal, serious or slight) for categorical variables. **A)** light conditions **B)** human controlled pedestrian crossing facilities **C)** physical pedestrian crossing facilities **D)** weather conditions

**A**

Graphical user interface, bar chart

Description automatically generated

**F**

**D**

**B**

**E**

**C**

**Figure 8:** Proportion of accident severities (fatal, serious or slight) for categorical variables. **A)** carriageway hazards **B)** high winds **C)** road surface conditions **D)** special conditions at site **E)** trunk road **F)** urban or rural area

> #make factor

> accident\_df$weather\_conditions<-as.factor(accident\_df$weather\_conditions)

> #change to weekday or not

> accident\_df$day\_of\_week <- as.integer(accident\_df$day\_of\_week)

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==1] <- "NO"

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==7] <- "NO"

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==2] <- "YES"

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==3] <- "YES"

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==4] <- "YES"

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==5] <- "YES"

> accident\_df$day\_of\_week[accident\_df$day\_of\_week==6] <- "YES"

> #make factor

> accident\_df$day\_of\_week <- as.factor(accident\_df$day\_of\_week)

> #check factor levels

> accident\_df %>% count(day\_of\_week)

day\_of\_week n

1 NO 65045

**Figure 9:** R code for converting ‘day\_of\_week’ from the day of the week into whether it was a weekday (‘YES’) or the weekend (‘NO’).

When converting the categorical attributes from numerical format to more easily interpreted factors, it was realised that some of the numbers corresponded to ‘unreported’ data and so was converted into missing values (NA). To check whether this had impacted the overall percentage of missing values, the missing data for each attribute was visualised once more [fig.10]. The number of missing data overall continued to be low (2.7%). However, the number of missing values for the class of the road where the accident first occurred (‘first\_road\_class’) was now 33.59%. Due to the high number of missing values and the possibility of other attributes, such as speed limit (‘speed\_limit’), road type (‘road\_type’) and whether the road was a trunk road or not (‘trunk\_road’), together potentially providing enough information to make up for the loss of road class, it was decided to remove ‘road\_class’ from the dataset. Additionally, the year, northing and easting coordinates, latitude and longitude were removed from the dataset as these variables are not useful for the association rules mining aims of this paper. The final dataset values can be seen in table 1.

A picture containing text, screenshot, antenna, cabinet

Description automatically generated

**Figure 10:** Plot of observations for each variable where black indicates observations where data is missing (NA).

**Table 1:** Final variables used in association rules mining with their corresponding values.

|  |  |
| --- | --- |
| **Numerical Attributes** | |
| Number of vehicles |  |
| Number of casualties |  |
| Hour | 0-24 |
| Month | January-December (1-12) |
| **Categorical Attributes** | |
| Weekday | YES |
| NO |
| Road type | Dual carriageway (DC) |
| One way street (OWS) |
| Roundabout (RB) |
| Single carriageway (SC) |
| Slip road (SR) |
| Speed limit | 20 |
| 30 |
| 40 |
| 50 |
| 60 |
| 70 |
| Junction Detail | More than 4 arms (4ARM) |
| Crossroads (CR) |
| Mini roundabout (MRB) |
| Private drive (PD) |
| Roundabout (RB) |
| Slip road (SR) |
| T-junction (TJUN) |
| Not at junction (NOT) |
| Pedestrian crossing – human controlled | YES |
| NO |
| Pedestrian crossing – physical facilities | YES |
| NO |
| Light conditions | Day |
| Lights – lit (LIT) |
| Lights – not lit (UNLIT) |
| No lights (NL) |
| Weather conditions | Fine |
| Fog |
| Rain |
| Snow |
| High winds | YES |
| NO |
| Road surface conditions | Dry |
| Flood |
| Ice |
| Snow |
| Wet |
| Special conditions at site | Auto traffic signal out (AFSO) |
| Auto traffic signal part defective (ASPD) |
| Road sign/marking obscured/defective (RSO) |
| Roadworks (RW) |
| Road surface defective (RSD) |
| Oil |
| Mud |
| None |
| Carriageway hazards | Vehicle load on road (LOAD) |
| Other object on road (OBJ) |
| Previous accident (ACC) |
| Animal in carriageway (ANI) |
| Pedestrian in carriageway (PED) |
| None |
| Urban or Rural area | Urban (URB) |
| Rural (RUR) |
| Trunk road | Trunk |
| Not |

3.2 Methodology

DTs are often used in classification problems, due to their simple structure, which when represented graphically is often visually easy to interpret (Tan et al., 2016). The class attribute of a new sample can be predicted by simply following a path from the root node to various other nodes until a terminal node (leaf) is reached. The leaf node is the class attribute assigned to a sample. To produce a DT an initial training dataset is supplied, which contains many records of attribute sets with a corresponding class attribute. An algorithm will then map the variables within the attribute set in order to be able to predict the class attribute. Often these algorithms grow trees recursively (Hunt’s Algorithm) (Tan et al., 2016). An initial test attribute is selected as a node (the root node). If the training records associated with the root node do not all belong to the same class, the data is split into purer subsets. The algorithm continues this way for each of the child nodes produced until the training records at the child node contain the same class, which produces a leaf node. There have been multiple DT methods developed which differ in the method of selecting a test attribute and the best split, for example, CART uses the Gini index and C4.5 uses information gain ratio (Tan et al., 2016). RFs are categorised as an ensemble learning method which generate multiple decision tree models and aggregate the results (Breiman, 2001). This makes them more robust to noise and outliers.

To assess the ability of a model to successfully predict the test datasets class, a confusion matrix is often used to count the number of records which are successfully classified verses those incorrectly classified (Tan et al., 2016). These counts can then be used to calculate the accuracy (the percentage of records classified correctly) and/or the opposite of this (the error rate). However, accuracy is not always the best performance metric to use as every class is equally considered (Tan et al., 2016). When dealing with imbalanced datasets where the class of interest may be rare, it may be better to treat each class as a binary classification, with a correct classification and a non-correct classification. The confusion matrix produced for each class would consist of true positive, true negative, false positive and false negative, which can then be used to calculate the precision and recall [table 2] (Tan et al., 2016). Precision measures the proportion of records which are true positive (correctly classified) out of all records which are classed as positive. Recall measures the proportion of records correctly classified. These measurements can then be combined into the F₁ measure to examine the trade-off between these two measurements. These measurements are calculated for each class so checks the model is accurately predicting all classes individually (Tan et al., 2016).

**Table 2:** Confusion matrix produced for binary classification. Can be used to calculate the precision and recall values.

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| 1 | 0 |
| 1 | True positive | False negative |
| 0 | False positive | True negative |

3.2.1 Implementation in R

The full R code for this section can be found in Appendix B. Initial exploration of the numeric attributes (number of vehicles, number of cars, hour and month) involved creating a table with the mean, standard deviation and median for each of the accident severities [table 3]. There does not appear to be any noticeable differences between the averages for any of the accident severities. Hour has a large standard deviation for each severity, which could be because of the peak in accidents around morning rush hour and evening rush hour [fig.4E]. This could be interesting used in the classification models.

**Table 3:** Summary statistics for numerical attributes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Accident Severity** | **Numerical Attribute** | **Count (n)** | **Mean** | **Standard Deviation** | **Median** |
| FATAL | Number of vehicles | 2624 | 1.78 | 0.99 | 2 |
| Number of casualties | 2624 | 1.71 | 1.24 | 1 |
| Hour | 2624 | 14.04 | 6.19 | 15 |
| Month | 2624 | 6.82 | 3.46 | 7 |
| SERIOUS | Number of vehicles | 35645 | 1.74 | 0.74 | 2 |
| Number of casualties | 35645 | 1.39 | 0.92 | 1 |
| Hour | 35645 | 14.60 | 5.35 | 15 |
| Month | 35645 | 6.66 | 3.36 | 7 |
| SLIGHT | Number of vehicles | 175885 | 1.88 | 0.70 | 2 |
| Number of casualties | 175885 | 1.31 | 0.73 | 1 |
| Hour | 175885 | 14.56 | 5.03 | 15 |
| Month | 175885 | 6.54 | 3.43 | 7 |
| TOTAL | Number of vehicles | 214154 | 1.86 | 0.72 | 2 |
| Number of casualties | 214154 | 1.33 | 0.77 | 1 |
| Hour | 214154 | 14.56 | 5.10 | 15 |
| Month | 214514 | 6.56 | 3.42 | 7 |

For the categorical attributes, the percentage of accidents categorised as each variable for the different accident severities was recorded [table 4]. Surprisingly, weather conditions, high winds and road surface conditions did not appear to differ largely between fatal accidents and non-fatal accidents. This is unlike previous findings which suggest weather plays an important role in the likelihood of road accidents (Das and Sun, 2014, Yu and Abdel-Aty, 2014). It does appear that more fatal accidents occur where there are not any streetlights, on roads above 60mph and not at a junction compared to non-fatal accidents. The most interesting observation is that fatal accidents appear much more likely to occur in rural areas whereas non-fatal accidents are more common in urban areas.

**Table 4:** Percentage of accidents which contain particular categorical attributes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Categorical Attribute** | | **FATAL**  **(%)** | **SERIOUS**  **(%)** | **SLIGHT**  **(%)** | **TOTAL**  **(%)** |
| Weekday | YES | 68.3 | 72.9 | 76.4 | 75.7 |
| NO | 31.7 | 27.1 | 23.6 | 24.3 |
| Road type | Dual carriageway (DC) | 20.9 | 13.8 | 16.0 | 15.7 |
| One way street (OWS) | 1.37 | 2.22 | 2.41 | 2.37 |
| Roundabout (RB) | 1.56 | 4.96 | 7.34 | 6.87 |
| Single carriageway (SC) | 75.4 | 78.3 | 73.1 | 74.0 |
| Slip road (SR) | 0.724 | 0.769 | 1.19 | 1.11 |
| Speed limit | 20 | 2.29 | 5.49 | 6.22 | 6.05 |
| 30 | 35.1 | 56.9 | 62.9 | 61.6 |
| 40 | 9.79 | 8.97 | 8.59 | 8.67 |
| 50 | 8.46 | 4.74 | 4.09 | 4.25 |
| 60 | 31.9 | 17.5 | 11.2 | 12.5 |
| 70 | 12.5 | 6.39 | 7.01 | 6.97 |
| Junction Detail | More than 4 arms (4ARM) | 0.229 | 0.735 | 0.988 | 0.936 |
| Crossroads (CR) | 5.98 | 8.91 | 10.5 | 10.2 |
| Mini roundabout (MRB) | 2.67 | 0.993 | 1.39 | 1.31 |
| Private drive (PD) | 2.48 | 3.14 | 2.93 | 2.96 |
| Roundabout (RB) | 2.74 | 6.28 | 9.46 | 8.85 |
| Slip road (SR) | 1.91 | 1.23 | 1.64 | 1.58 |
| T-junction (TJUN) | 20.8 | 32.4 | 32.6 | 32.4 |
| Not at junction (NOT) | 65.6 | 46.3 | 40.5 | 41.8 |
| Pedestrian crossing – human controlled | YES | 0.381 | 0.853 | 0.998 | 0.967 |
| NO | 99.6 | 99.1 | 99.0 | 99.0 |
| Pedestrian crossing – physical facilities | YES | 14.6 | 17.9 | 19.8 | 19.4 |
| NO | 85.4 | 82.1 | 80.2 | 80.6 |
| Light conditions | Day | 58.8 | 70.3 | 74.2 | 73.3 |
| Lights – lit (LIT) | 21.2 | 22.0 | 20.7 | 20.9 |
| Lights – not lit (UNLIT) | 0.877 | 0.746 | 24.2 | 0.644 |
| No lights (NL) | 19.1 | 7.00 | 4.53 | 5.12 |
| Weather conditions | Fine | 88.1 | 87.9 | 87.1 | 87.2 |
| Fog | 0.915 | 0.609 | 0.546 | 0.561 |
| Rain | 10.6 | 11.1 | 12.0 | 11.8 |
| Snow | 0.343 | 0.339 | 0.408 | 0.396 |
| High winds | YES | 2.71 | 1.96 | 1.89 | 1.91 |
| NO | 97.3 | 98.0 | 98.1 | 98.1 |
| Road surface conditions | Dry | 70.2 | 74.3 | 74.2 | 74.1 |
| Flood | 0.114 | 0.0898 | 0.0927 | 0.0925 |
| Ice | 1.33 | 1.14 | 1.30 | 1.27 |
| Snow | 0.191 | 0.157 | 0.219 | 0.209 |
| Wet | 28.2 | 24.3 | 24.2 | 24.3 |
| Special conditions at site | Auto traffic signal out (AFSO) | 0.152 | 0.188 | 0.219 | 0.213 |
| Auto traffic signal part defective (ASPD) | 0.000 | 0.0224 | 0.0409 | 0.0374 |
| Road sign/marking obscured/defective (RSO) | 0.152 | 0.160 | 0.113 | 0.121 |
| Roadworks (RW) | 0.991 | 0.864 | 1.08 | 1.04 |
| Road surface defective (RSD) | 0.152 | 0.387 | 0.147 | 0.187 |
| Oil | 0.114 | 0.300 | 0.186 | 0.205 |
| Mud | 0.114 | 0.267 | 0.248 | 0.250 |
| None | 98.3 | 97.8 | 98.0 | 97.9 |
| Carriageway hazards | Vehicle load on road (LOAD) | 0.191 | 0.174 | 0.171 | 0.172 |
| Other object on road (OBJ) | 1.14 | 0.957 | 0.752 | 0.791 |
| Previous accident (ACC) | 0.686 | 0.146 | 0.126 | 0.136 |
| Animal in carriageway (ANI) | 0.152 | 0.424 | 0.366 | 0.373 |
| Pedestrian in carriageway (PED) | 0.152 | 0.199 | 0.185 | 0.187 |
| None | 97.7 | 98.1 | 98.4 | 98.3 |
| Urban or Rural area | Urban (URB) | 37.8 | 60.2 | 67.9 | 66.2 |
| Rural (RUR) | 62.2 | 39.8 | 32.1 | 33.8 |
| Trunk road | Trunk | 14.6 | 7.37 | 8.35 | 8.27 |
| Not | 85.4 | 92.6 | 91.6 | 91.7 |

The dataset was split into a training and test set using the function ‘initial\_split’ from the package ‘rsample’ (Silge et al., 2021) using the proportion 0.7. The proportion was selected due to most other research into using classification modelling to predict accident severity used 70% of the data for training and the remaining 30% for testing their model. The function uses random sampling to select the records assigned to either testing or training. In this type of sampling there is equal probability of selecting any record. To ensure the proportion of each accident severity class represented in both the training and testing set is the same as the proportion in the original dataset, the argument ‘strata’ was set equal to ‘accident\_severity’, allowing stratified sampling [table 5].

**Table 5:** Number of samples for each accident severity in each dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Accident severity** | **Original dataset** | **Training dataset** | **Testing dataset** |
| Fatal | 2624 | 1828 | 796 |
| Serious | 35645 | 24949 | 10696 |
| Slight | 175885 | 123130 | 52755 |

To produce the decision tree model, the method used came from the ‘ctree’ function (Hothorn et al., 2006) in the ‘party’ package. ‘ctree’ was selected as it produces Conditional Inference Trees, which select test attributes and best splits using permutation tests (multiple significance tests) (Hothorn et al., 2006). Each expansion of the tree needs to produce a significant improvement to the model. This method was designed to avoid the selection bias towards attributes with many possible splits that other common algorithms are prone to, such as the popular CART algorithm. The selection bias can then affect tree interpretation, potentially affecting the accuracy of the model. A common problem with DTs is the tendency to overfit (Tan et al., 2016). Overfitting is where DTs fit the training dataset so well, that the model is then not generalised enough to predict the class of the testing dataset accurately and so is not useful for prediction of real data. This is especially true for large DTs, which is likely to be produced by the dataset in this study due to high dimensionality. To prevent overfitting cross-validation using 10-folds was included as a control. The training dataset is split into 10 equally sized partitions, with one of the partitions being used as a testing set and the others combined into a training set for each run. Each partition is used as the testing set once and the accuracy for each produced model is calculated. This allows the selection of the most optimal tree (Tan et al., 2016). The package ‘caret’ is an excellent package in R for building models due to its comprehensive framework (Kuhn, 2021). The ‘train’ function can apply a variety of different model-building functions in R from different packages and combine these with their own control features to produce optimal tuning values in one line of code [fig.11]. The mincriterion value that produces the optimal tree size, in terms of accuracy and avoiding overfitting, is produced by this function [fig.11]. The suggested mincriterion value was then used as a control for the ‘ctree’ function to build the model.

> control <- trainControl(method = "cv", number = 10)

> accident\_dt <- train(accident\_severity~., accident\_train, method="ctree",

+ trControl= control)

> accident\_dt

Conditional Inference Tree

149907 samples

18 predictor

3 classes: 'FATAL', 'SERIOUS', 'SLIGHT'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 134916, 134917, 134916, 134916, 134916, 134916, ...

Resampling results across tuning parameters:

mincriterion Accuracy Kappa

0.01 0.8203753 0.009077713

0.50 0.8213025 0.007824815

0.99 0.8213893 0.005666622

Accuracy was used to select the optimal model using the

largest value.

The final value used for the model was mincriterion = 0.99.

**Figure 11:** caret package output for optimal decision tree.

The produced decision tree could then be plotted visually to better interpret the relationship between the accident characteristics and the class severity outcome [fig.12]. The resulting tree was very large, as can be seen in figure 12. The model was applied to the test set and a confusion matrix was produced [table 6]. The overall model accuracy was 0.821 (95% confidence interval: 0.818, 0.8239) [table 7]. Looking at accuracy alone, this model appears to be very good at predicting the accident severity of a crash. However, considering the confusion matrix [table 6] and the other statistics measures produced for each class [table 7], this only appears to be true for slight accidents. This is likely due to the large class imbalance, with 82% of the records belonging having an accident severity of slight. Predicting the accident as slight most of the time is a strategy that will work out correct most of the time, however, is not useful in a practical sense. It is more useful to understand and predict fatal and serious accidents if we intend to use the information to reduce road-related deaths and injury.

A picture containing text, nature

Description automatically generated

**Figure 12:** Decision tree classifying accident severity.

**Table 6:** Confusion matrix produced for decision tree classification of accident severity performance.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 0 | 1 | 795 |
| SERIOUS | 0 | 8 | 10688 |
| SLIGHT | 0 | 19 | 52736 |

**Table 7:** Performance metrics for decision tree performance in classifying accident severity.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Class** | | |
| **Fatal** | **Serious** | **Slight** |
| **Precision** | NA | 0.2857143 | 0.8211900 |
| **Recall** | 0.00000 | 0.0007479 | 0.9996398 |
| **F1** | NA | 0.0014920 | 0.9016705 |

To improve the model’s ability to predict rarer classes, processes call downsampling and oversampling can be applied to the dataset. Downsampling involves making the number of records for each class equal to the rarest class (slight); however, this could lead to loss of useful data from the other classes. Oversampling has been shown to be a better solution, as it involves replicating the records in smaller classes (fatal and serious) until they are equal to the largest class (slight) and so potentially useful data is not lost (Savolainen et al., 2011). The best solution is to combine both of these methods. However, due to the size of the dataset and lack of computing power, for this study it was only possible to use downsampling. The training dataset was downsampled using the function ‘downSample’ from the ‘caret’ package (Kuhn, 2021) so each accident severity class contained 1828 records. The same modelling process was replicated for the downsampled training set. The optimal mincriterion for the downsampled dataset was 0.5. This produced a tree which was also very large with a reduced accuracy of (0.513; 95% confidence interval: 0.5092, 0.5169) [table 9]. However, this tree had an improved ability to detect fatal and serious accidents [table 8]. One of the benefits of using DT is the easy visual interpretation of the resulting tree [fig.13]. As this produced tree was fairly large, it was more difficult to interpret. To simplify the tree, a max depth was set as 5. This made the tree much more visibly easy to interpret and the resulting accuracy was 0.505 (95% confidence interval: 0.5012, 0.5089). This smaller tree was kept as the reduction in accuracy was worth the trade off to better interpret the variables which contribute to accident severity.

Chart

Description automatically generated

**Figure 13:** Decision tree classifying accident severity using the downsampled dataset.

**Table 8:** Confusion matrix produced for decision tree classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 538 | 80 | 178 |
| SERIOUS | 4573 | 1548 | 6415 |
| SLIGHT | 15465 | 4575 | 30875 |

**Table 9:** Performance metrics for decision tree performance in classifying accident severity using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Class** | | |
| **Fatal** | **Serious** | **Slight** |
| **Precision** | 0.026147 | 0.19247 | 0.8666 |
| **Recall** | 0.675879 | 0.14473 | 0.5853 |
| **F1** | 0.050346 | 0.16522 | 0.6987 |

To ensure the RF model is also not affected by class imbalance, the downsampled training dataset used for producing the DT model was used. The ‘train’ function from the ‘caret’ package (Kuhn, 2021) was applied to gain the mtry value which produced the highest accuracy [fig.14]. The mtry value is the number of attributes randomly sampled as candidate variables at each split. Although random forest tends to avoid overfitting due to bagging, the same 10-fold cross validation control as used in DT was included as an extra control. The resulting mtry of 2 was then used as the control in the function ‘randomForest’ from the ‘randomForest’ package (Liaw and Wiener, 2002) to produce the optimal RF model. The maximum number of trees was 500. The error rate seems to have levelled off before 500 trees, however, there still appears to be some variation so the maximum number of trees was left at 500 [fig.15].

> accident\_rf

Random Forest

5484 samples

18 predictor

3 classes: 'FATAL', 'SERIOUS', 'SLIGHT'

No pre-processing

Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 4935, 4935, 4935, 4936, 4937, 4935, ...

Resampling results across tuning parameters:

mtry Accuracy Kappa

2 0.4626308 0.1939523

25 0.4482220 0.1723074

48 0.4383656 0.1575253

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was mtry = 2.

**Figure 14:** caret package output for optimal random forest model.

Chart

Description automatically generated with medium confidence

**Figure 15:** The error for the random forest model output plotted against the number of trees used.

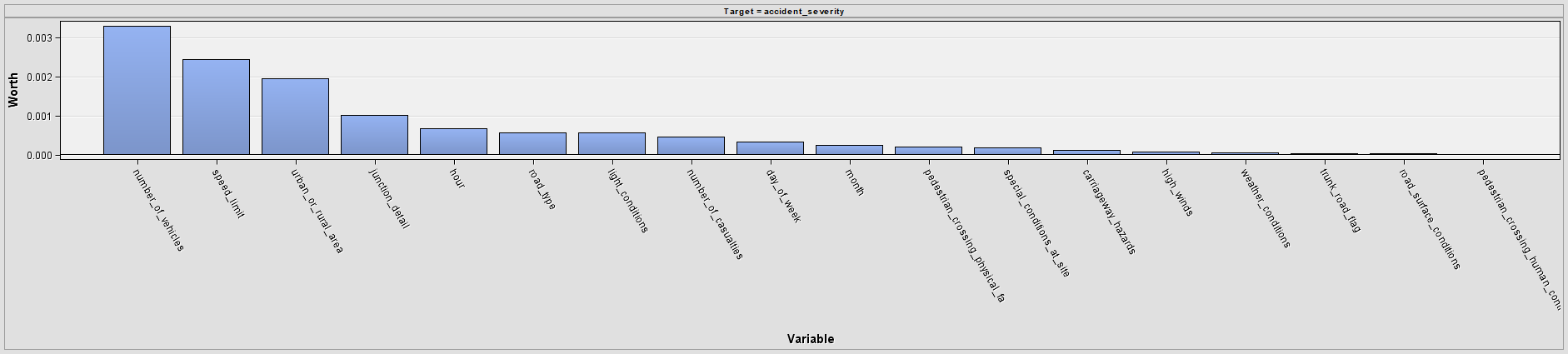
3.2.2 Implementation in SAS

The prepared dataset was imported into SAS EM using the ‘File Import’ node, with the role ‘Train’. For classification in SAS EM a variable must be given the role of ‘Target’ as the assigned class attribute; this was set as the accident severity. Data exploration using the ‘StatExplore’ node provided the same results as those in table and table produced using R. However, SAS EM produces additional charts which display the Chi-square value for [fig.16] and worth of each variable [fig.17]. These appear to suggest that urban or rural area, speed limit, junction detail, road type, number of vehicles, day of week, number of casualties and light conditions are important factors for accident severity classification.

Chart

Description automatically generated

**Figure 16:** Chi-squared value for each of the input variables with accident severity set as the target.



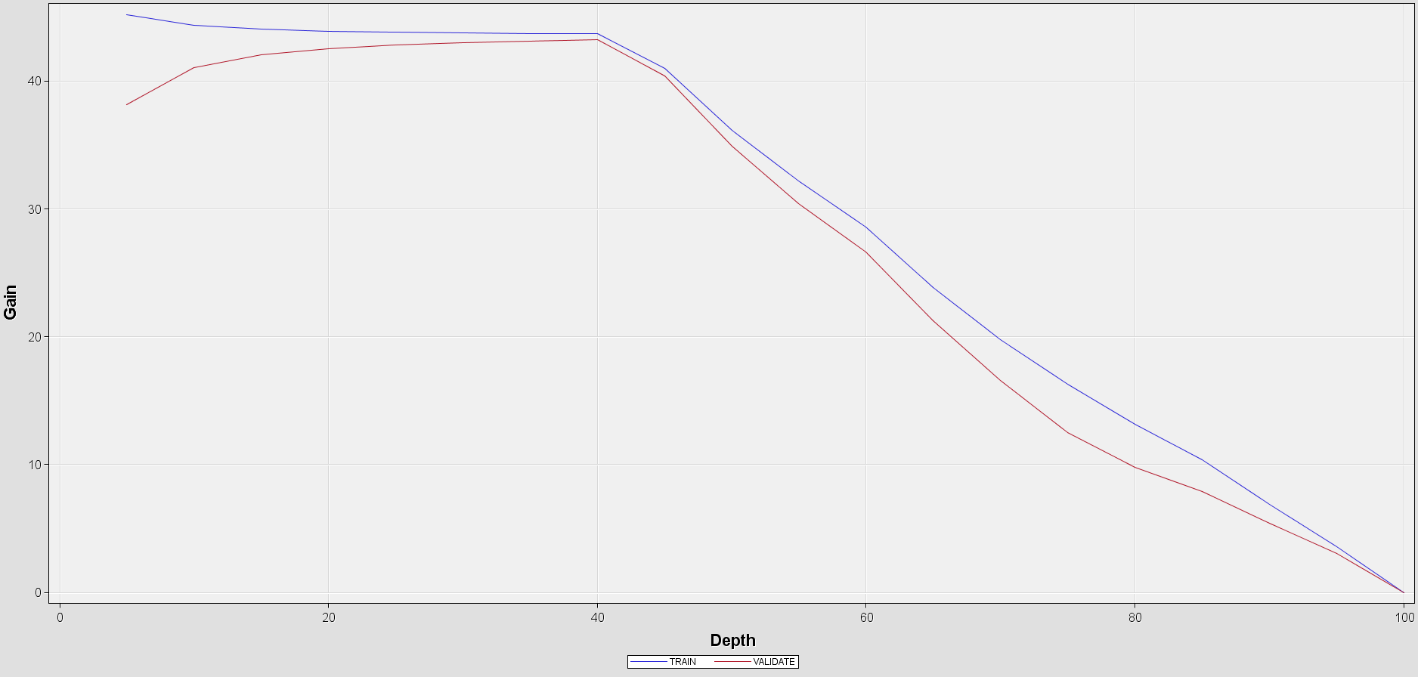
**Figure 17:** Worth of each of the input variables with accident severity set as the target.

When partitioning the dataset, SAS EM includes the option of creating a validation set and test set, along with the training set. The validation set is used to tune the hyperparameters of the model and the test set then obtains an estimate of the model’s performance on unseen data. The values for partitioning were set as 70% allocated to training, 20% for validation and 10% for testing. The ‘HP tree’ node was used to create the decision tree as classification was not binary for this dataset, whereas the ‘Decision Tree’ node appeared only to allow for binary classification. The Gini index was used to determine splitting. This was chosen as the Gini index is used by the CART algorithm, which has been a popular choice when modelling accident severity, so it will be interesting to compare its performance to the conditional inference trees used in the R implementation. Maximum branch and maximum depth were left as 2 and 10, respectively. Similar to R the tree produced was very large and was also affected by class imbalance, with majority of accidents assigned the severity slight and none predicted as fatal in the validation dataset [table 10], despite the test set having a misclassification rate of 17.7% (an accuracy of 82.3%).

**Table 10:** Confusion matrix produced for decision tree classification of accident severity performance in SAS EM.

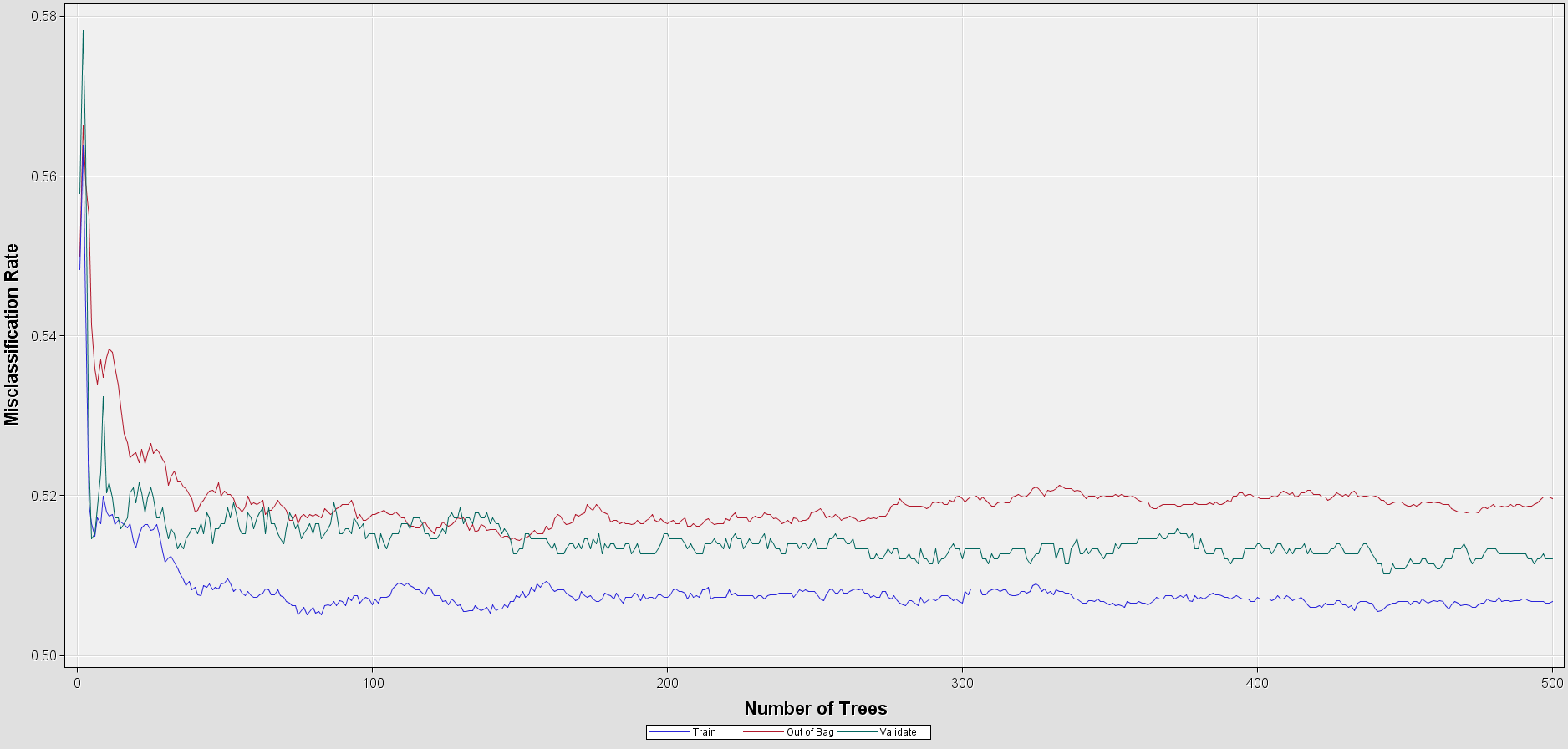
|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 0 | 5 | 520 |
| SERIOUS | 0 | 8 | 7121 |
| SLIGHT | 0 | 20 | 35155 |

As in the R implementation, we are more interested in predicting fatal and serious accidents compared to slight, so using the ‘Sample’ node the dataset was downsampled so each accident severity contained 2624 samples. Despite efforts, I was unable to use the ‘Sample’ node to upsample or alter the amount of both fatal and serious accidents, so they held more weight. These alternative methods would have been preferred as the number of samples had been much reduced, meaning potential information may have been lost, as in the R implementation. The data was partitioned, and the decision tree initially created using the same parameters as first tree produced in SAS EM, producing a large decision tree with a misclassification rate of 54.9% for the test set. Reducing the maximum depth to 5 made the tree much smaller and produced a misclassification rate of 54.2% in the test set, so this tree was preferred. Additionally, increasing the depth of the tree does appear to increase the gain of the validation set, but not by much [fig.18]. As the benefit of using DTs is their easy interpretability, the increased interpretability into the contributing factors to accident severity outweighs the increased gain from increasing the depth of the tree.

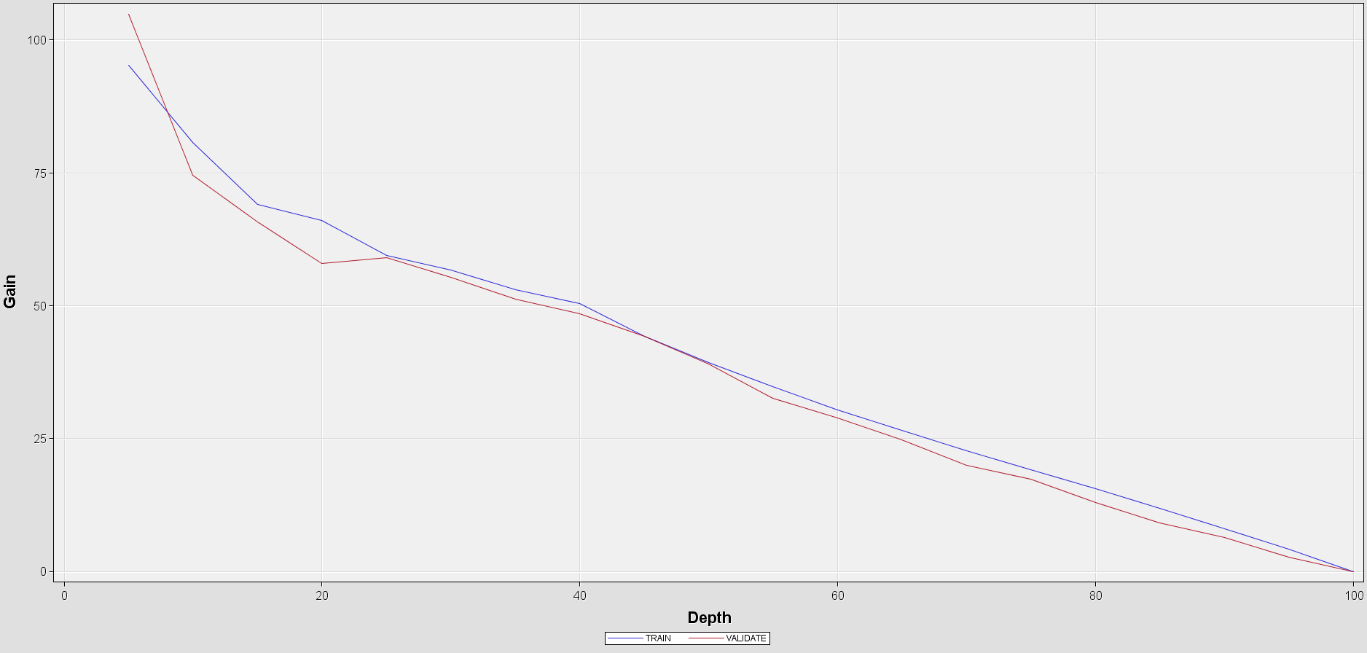


**Figure 18:** Gain plotted against decision tree depth for the training and validation dataset.

RF modelling used the same downsampled dataset as the DT to prevent class imbalance from having an effect. The node ‘HP forest’ was used and the maximum number of trees was 500 with a maximum depth of 50. There still appeared to be some variation in the misclassification rate around 500 trees in the validation set, however even increasing to 800 did not seem to reduce this level of variation so the model was left at maximum 500 trees [fig.19]. The gain for the validation set seemed to drop off quickly as maximum depth increased but increased at a depth of 25 to become more stable and a similar gain to the training set, so the maximum depth was then changed to 25 [fig.20].



**Figure 19:** The misclassification rate of the random forest model in classifying accident severity according to the number of trees used.



**Figure 20:** Gain plotted against random forest tree depth for the training and validation dataset.

The full SAS EM pathway for DT and RF model development is shown in figure 21.

Diagram

Description automatically generated

**Figure 21:** Diagram showing SAS EM workflow for production of decision trees and random forest model.

4. Results analysis and discussion

**Decision Tree**

The accuracy of the DT produced by R was 0.505 (95% confidence interval: 0.5012, 0.5089) [table 11]. Although the accuracy was reduced overall, the F₁ -score for fatal accidents was higher at 0.0541 [table 12] for the tree with reduced depth than the maximum depth tree (F₁ =0.0503). The tree has 22 terminal nodes and the speed limit, number of vehicles, junction detail, number of casualties, urban/rural area, light conditions, month, hour, and road type [fig.22]. This would suggest these variables are the most important when it comes to determining accident injury severity. However, the tree is not simple to interpret, as it is not clear that one section of the tree contains more of one type of accident. There is some indication that that higher speed limits (>50mph) and not being at a junction is more likely to lead to fatal accidents. Additionally, the number of casualties appears to relate to whether an accident is more likely to be only slight or more serious/fatal, with more than one casualty being associated with higher severity accidents. Unfortunately, the difference in the number allocated to each accident severity at some nodes does not appear significantly different (e.g., node 22, node 30, node 6) and so the different severities would be nearly equally likely to be allocated to this node. Additionally, it does not appear that serious accidents can be easily distinguished and classified, with only node 26 having more allocations of serious accidents than the other severities.

Diagram

Description automatically generated with medium confidence

**Figure 22:** Decision tree classifying accident severity using the downsampled dataset with a maximum depth of 5 produced by R.

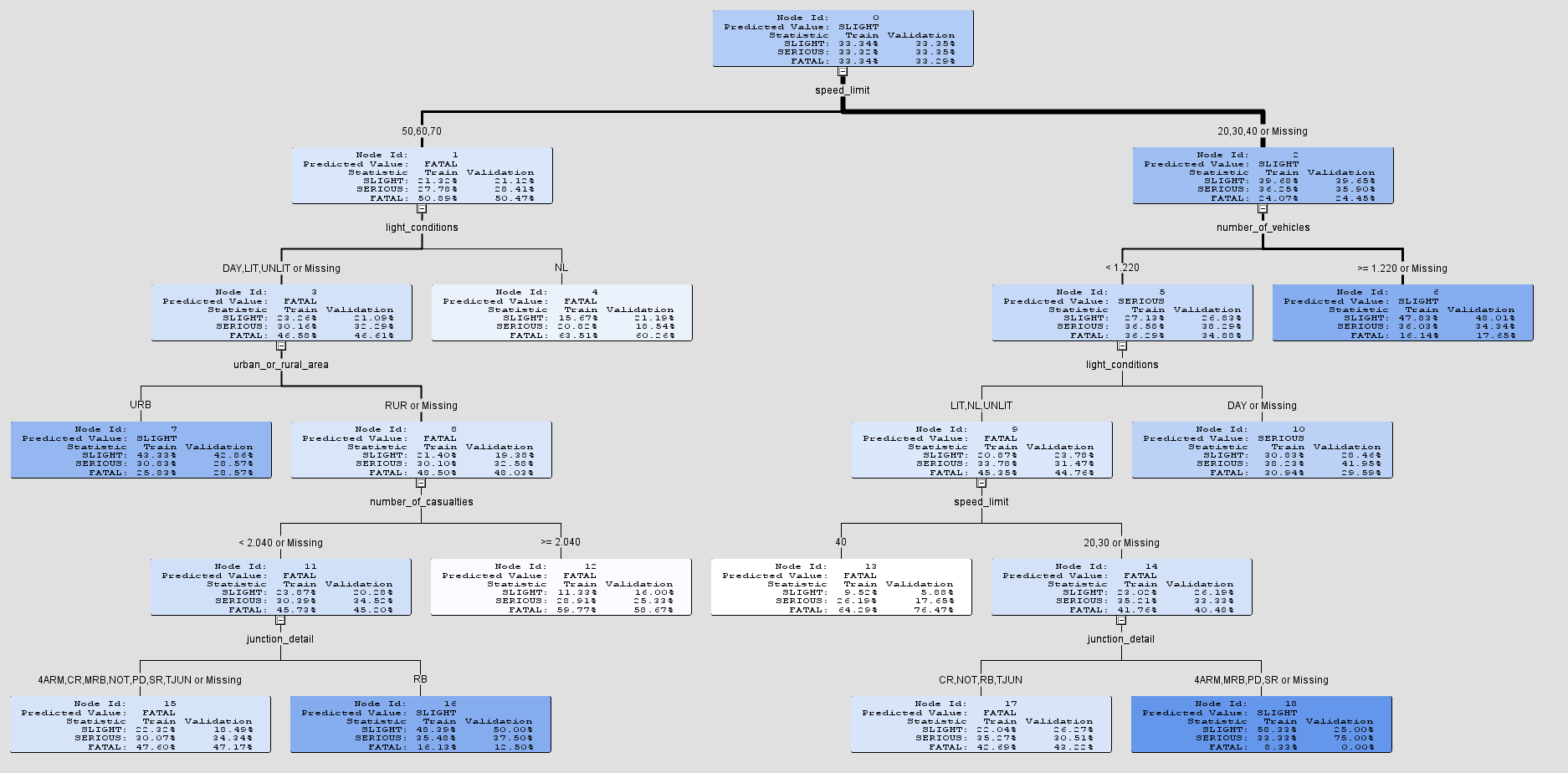
**Table 11:** Confusion matrix produced for R produced decision tree classification of accident severity performance using the downsampled dataset with maximum tree depth of 5.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 520 | 109 | 167 |
| SERIOUS | 4136 | 2235 | 4325 |
| SLIGHT | 13762 | 9300 | 29693 |

**Table 12:** Performance metrics for R produced decision tree performance in classifying accident severity using the downsampled dataset with maximum tree depth of 5.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Class** | | |
| **Fatal** | **Serious** | **Slight** |
| Precision | 0.0282 | 0.1919 | 0.8686 |
| Recall | 0.6533 | 0.2090 | 0.5628 |
| F₁ | 0.0541 | 0.2001 | 0.6831 |

The DT produced by SAS EM had a misclassification rate of 0.5444 (so an accuracy of 0.4556) [table 13 & table 15], which is lower than the DT model produced by R. However, the validation set for SAS EM had a much higher F₁ -score (0.5662) for fatal accidents [table 14] compared to the R DT model. The tree had 9 terminal nodes, which is much less than tree produced by R [fig.23]. The variables used were speed limit, number of vehicles, light conditions, urban/rural, number of casualties, junction detail, which is very similar to those used in R except SAS EM did not use month, hour and road type. Although slightly easy to interpret due to the SAS EM tree having fewer determining variables, it is still not straightforward. Speed limit also produces the first split, with more fatal accidents being associated with higher speeds (>50). It appears some of the most influential factors for fatal accidents are higher speeds with low light levels or in rural areas. Unfortunately, once more serious is not well represented in this tree, with only one terminal node predicting accidents as serious [fig.23], even though the proportion of serious accidents at that node is not greatly higher than the proportion of slight and fatal accidents.



**Figure 23:** Decision tree classifying accident severity using the downsampled dataset produced by SAS EM.

**Table 13:** Confusion matrix produced for SAS EM produced decision tree classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 340 | 79 | 105 |
| SERIOUS | 195 | 115 | 215 |
| SLIGHT | 142 | 77 | 306 |

**Table 14:** Performance metrics for SAS EM produced decision tree performance in classifying accident severity using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Class** | | |
| **Fatal** | **Serious** | **Slight** |
| **Precision** | 0.5022 | 0.4244 | 0.5840 |
| **Recall** | 0.6489 | 0.2190 | 0.5829 |
| **F₁** | 0.5662 | 0.2890 | 0.5834 |

**Table 15:** Fit statistics for SAS EM produced decision tree performance in classifying accident severity using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Fit statistics** | | |
| Train | Validation | Test |
| Average squared error | 0.2047 | 0.2072 | 0.2088 |
| Maximum absolute error | 0.8867 | 0.8867 | 0.8867 |
| Misclassification rate | 0.5156 | 0.5156 | 0.5444 |

**Random Forest**

The random forest model produced by R had an accuracy of 0.5208 (95% confidence interval: 0.5170, 0.5247) [table 16 & table 17], which is higher than the DT produced by R. The model was still much better at classifying slight accidents compared to the lower frequency accident severities (serious and slight) [table 16]. The F₁-score for predicting fatal accidents was very low at 0.0574 [table 17]. A random forest model is more difficult to interpret in terms of visualising how different variables interact to produce certain accident injury severity outcomes compared to a DT. It is still possible to see the most important variables for influencing accident severity, although not necessarily which ones are more important for particular severities. The top five important variables picked up by the RF model were hour, speed limit, month, number of vehicles and junction detail [fig.24]. These variables are also included in the DT produced by R as some of the most influential factors for accident severity.

Chart, scatter chart

Description automatically generated

**Figure 24:** Importance of each input for classifying accident severity determined by the random forest model produced by R.

**Table 16:** Confusion matrix produced for R produced random forest classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 522 | 103 | 12936 |
| SERIOUS | 3939 | 2346 | 4411 |
| SLIGHT | 12936 | 9224 | 30595 |

**Table 17:** Performance metrics for R produced random forest classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Class** | | |
| **Fatal** | **Serious** | **Slight** |
| **Precision** | 0.0300 | 0.2010 | 0.8697 |
| **Recall** | 0.6558 | 0.2193 | 0.5799 |
| **F**₁ | 0.0574 | 0.2098 | 0.6959 |

The RF model produced by SAS EM had an accuracy of 0.4936 (misclassification rate of 0.5064) [table 18 & table 20] which is higher than the DT produced using SAS EM but is lower than the RF model produced by R. However, the F₁-score for predicting fatal accidents was much higher at 0.5652 [table 19] than the RF model produced by R. The top five most influential factors for accident severity in this model are number of vehicles, speed limit, junction detail, urban or rural area and light conditions [fig.25]. These are slightly different to the previous RF model, but are the same variables used in the SAS EM produced DT to classify accident severity.

Chart

Description automatically generated

**Figure 25:** Importance of each input for classifying accident severity determined by the random forest model produced by SAS EM.

**Table 18:** Confusion matrix produced for SAS EM produced random forest classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | **Predicted** | | |
| FATAL | SERIOUS | SLIGHT |
| FATAL | 323 | 104 | 97 |
| SERIOUS | 174 | 142 | 209 |
| SLIGHT | 122 | 99 | 304 |

**Table 19:** Performance metrics for SAS EM produced random forest classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance metric** | **Class** | | |
| **Fatal** | **Serious** | **Slight** |
| **Precision** | 0.5218 | 0.4116 | 0.4984 |
| **Recall** | 0.6164 | 0.2705 | 0.5790 |
| **F**₁ | 0.5652 | 0.3265 | 0.5355 |

**Table 20:** Fit statistics for SAS EM produced random forest classification of accident severity performance using the downsampled dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fit Statistic** | **Dataset** | | |
| Train | Validation | Test |
| Average squared error | 0.2026 | 0.2050 | 0.2046 |
| Maximum absolute error | 0.8538 | 0.8728 | 0.8474 |
| Misclassification rate | 0.5064 | 0.5114 | 0.5228 |

5. Conclusion

The accuracy produced by both R models were higher than those produced using SAS EM. However, the SAS EM model seemed better able to predict the lower frequency severities, especially fatal accidents. In both instances the random forest models produced higher accuracy compared to the decision tree. This has been found by other studies comparing model performance for predicting accident severity (Krishnaveni and Hemalatha, 2011, Mafi et al., 2018). It is likely because random forest models aggregate multiple decision trees. However, the random forest models both had an accuracy of 52.08% and 49.36%. Although similar to the accuracy found by other research into classifying accident severity (de Oña et al., 2013, Krishnaveni and Hemalatha, 2011), this level of accuracy is not useful for production of a model to predict accident severity in live time. Additionally, the benefit of using a decision tree is the ability to visualise how the factors interact to produce the accident severity (Silva et al., 2020). However, the decision trees produced were not simple to interpret. Instead, it may be more useful to consider the outputs of variable importance as worth further investigation into the causes of accidents and how to reduce the likelihood of future crashes.

The prioritised factors used by the models produced by SAS EM and R differed slightly. This is likely due to the models using different selection criteria when splitting the tree, with R using conditional inference trees and SAS EM using the Gini index. Another study found that when classifying accident severity, different models prioritised different factors (Zhang et al., 2018). The common variables were light conditions, urban or rural area, speed limit and junction detail. In earlier data exploration, these variables appeared to produce different proportions of accident severity so appeared to have some influence, particularly on fatal accidents. Other research has also identified environmental conditions, such as lighting, road type and speed limit, as highly influential in determining accident severity (Zeng and Huang, 2014, De Oña et al., 2011, Feng et al., 2019). Many studies have also emphasised the importance of human factors for influencing accident severity, such as wearing seatbelts or driver age (Xu et al., 2018, Tavakoli Kashani et al., 2011, Pakgohar et al., 2011). In the future, it may be interesting to repeat these models including driver and vehicle information to see if this produces improved accuracy.

6. References

AMIRI, A. M., SADRI, A., NADIMI, N. & SHAMS, M. 2020. A comparison between artificial neural network and hybrid intelligent genetic algorithm in predicting the severity of fixed object crashes among elderly drivers. *Accident Analysis & Prevention,* 138**,** 105468.

BREIMAN, L. 2001. Random forests. *Machine learning,* 45**,** 5-32.

CHANG, L.-Y. & WANG, H.-W. 2006. Analysis of traffic injury severity: An application of non-parametric classification tree techniques. *Accident Analysis & Prevention,* 38**,** 1019-1027.

DAS, S. & SUN, X. Investigating the pattern of traffic crashes under rainy weather by association rules in data mining. Transportation Research Board 93rd Annual Meeting, 2014. Transportation Research Board Washington, DC.

DE OÑA, J., LÓPEZ, G. & ABELLÁN, J. 2013. Extracting decision rules from police accident reports through decision trees. *Accident Analysis & Prevention,* 50**,** 1151-1160.

DE OÑA, J., MUJALLI, R. O. & CALVO, F. J. 2011. Analysis of traffic accident injury severity on Spanish rural highways using Bayesian networks. *Accident Analysis & Prevention,* 43**,** 402-411.

DELEN, D., TOMAK, L., TOPUZ, K. & ERYARSOY, E. 2017. Investigating injury severity risk factors in automobile crashes with predictive analytics and sensitivity analysis methods. *Journal of Transport & Health,* 4**,** 118-131.

DEPARTMENT FOR PUBLIC TRANSPORT 2021a. Road Safety - Accidents last 5 years.

DEPARTMENT FOR PUBLIC TRANSPORT 2021b. Road Safety Open Dataset Data Guide.

DEPARTMENT FOR TRANSPORT 2021. Reported road casualties in Great Britain, provisional estimates: year ending June 2021.

FENG, M., ZHENG, J., REN, J. & XI, Y. Association Rule Mining for Road Traffic Accident Analysis: A Case Study from UK. International Conference on Brain Inspired Cognitive Systems, 2019. Springer, 520-529.

HOTHORN, T., HORNIK, K. & ZEILEIS, A. 2006. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics,* 15**,** 651-674.

IRANITALAB, A. & KHATTAK, A. 2017. Comparison of four statistical and machine learning methods for crash severity prediction. *Accident Analysis & Prevention,* 108**,** 27-36.

KRISHNAVENI, S. & HEMALATHA, M. 2011. A perspective analysis of traffic accident using data mining techniques. *International Journal of Computer Applications,* 23**,** 40-48.

KUHN, M. 2021. caret: Classification and Regression Training.

LI, Z., LIU, P., WANG, W. & XU, C. 2012. Using support vector machine models for crash injury severity analysis. *Accident Analysis & Prevention,* 45**,** 478-486.

LIAW, A. & WIENER, M. 2002. Classification and Regression by randomForest. *R News,* 2**,** 18-22.

MAFI, S., ABDELRAZIG, Y. & DOCZY, R. 2018. Machine learning methods to analyze injury severity of drivers from different age and gender groups. *Transportation research record,* 2672**,** 171-183.

PAKGOHAR, A., TABRIZI, R. S., KHALILI, M. & ESMAEILI, A. 2011. The role of human factor in incidence and severity of road crashes based on the CART and LR regression: a data mining approach. *Procedia Computer Science,* 3**,** 764-769.

R CORE TEAM 2021. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

SAS INSTITUTE INC. 2015. SAS® Enterprise Miner™ 14.1: Administration and

Configuration. Cary, NC: SAS Institute Inc.

SAVOLAINEN, P. T., MANNERING, F. L., LORD, D. & QUDDUS, M. A. 2011. The statistical analysis of highway crash-injury severities: a review and assessment of methodological alternatives. *Accident Analysis & Prevention,* 43**,** 1666-1676.

SILGE, J. 2021. Class imbalance and classification metrics with aircraft wildlife strikes. *Julia Silge* [Online]. Available from: https://juliasilge.com/blog/sliced-aircraft/.

SILGE, J., CHOW, F., KUHN, M. & WICKHAM, H. 2021. rsample: General Resampling Infrastructure.

SILVA, P. B., ANDRADE, M. & FERREIRA, S. 2020. Machine learning applied to road safety modeling: A systematic literature review. *Journal of Traffic and Transportation Engineering (English Edition),* 7**,** 775-790.

TAN, P.-N., STEINBACH, M. & KUMAR, V. 2016. *Introduction to data mining*, Pearson Education India.

TAVAKOLI KASHANI, A., SHARIAT-MOHAYMANY, A. & RANJBARI, A. 2011. A data mining approach to identify key factors of traffic injury severity. *PROMET-Traffic&Transportation,* 23**,** 11-17.

TIERNEY, N. 2017. visdat: Visualising Whole Data Frames. *Journal of Open Source Software,* 2**,** 355.

WAHAB, L. & JIANG, H. 2019. A comparative study on machine learning based algorithms for prediction of motorcycle crash severity. *PLOS ONE,* 14**,** e0214966.

WEI, T. & SIMKO, V. 2021. R package 'corrplot': Visualization of a Correlation Matrix.

WEN, X., XIE, Y., JIANG, L., PU, Z. & GE, T. 2021. Applications of machine learning methods in traffic crash severity modelling: current status and future directions. *Transport reviews,* 41**,** 855-879.

WORLD HEALTH ORGANIZATION 2018. Global status report on road safety 2018. .

XU, C., BAO, J., WANG, C. & LIU, P. 2018. Association rule analysis of factors contributing to extraordinarily severe traffic crashes in China. *Journal of safety research,* 67**,** 65-75.

YU, R. & ABDEL-ATY, M. 2014. Analyzing crash injury severity for a mountainous freeway incorporating real-time traffic and weather data. *Safety science,* 63**,** 50-56.

ZENG, Q. & HUANG, H. 2014. A stable and optimized neural network model for crash injury severity prediction. *Accident Analysis & Prevention,* 73**,** 351-358.

ZHANG, J., LI, Z., PU, Z. & XU, C. 2018. Comparing Prediction Performance for Crash Injury Severity Among Various Machine Learning and Statistical Methods. *IEEE Access,* 6**,** 60079-60087.

7. Appendix

**Appendix A**

R code used for data preparation and exploration.

#DATA EXPLORATION & PREP

library(tidyverse)

library(ggplot2)

library(Hmisc)

library(visdat)

rm(list=ls())

#set working directory

setwd("C:\\Users\\smann\\OneDrive\\University\\Data Science\\ASDM\\Project\\Road safety")

#ACCIDENT DATASET

df <- read.csv("dft-road-casualty-statistics-accident-last-5-years.csv")

#exploration

head(accident\_df)

tail(accident\_df)

summary(accident\_df)

str(accident\_df)

names(accident\_df)

dim(accident\_df)

#dataset large so selecting specific years to reduce size - selected 2016-2018

accident\_df <- df %>% filter(accident\_year<2018)

#change ï..accident\_index to accident\_index

accident\_df <- accident\_df %>% rename(accident\_index=ï..accident\_index)

#check for duplicated data - no duplicates

table(duplicated(accident\_df$accident\_index))

#remove initally selected unnecessary variables

#remove accident reference, police force, local\_authority\_district,

#local\_authority\_ons\_district, local\_authority\_highway, first\_road\_number,

#second\_road\_number, lsoa\_of\_accident\_location, did\_police\_officer\_attend\_scene\_of\_accident

accident\_df <- accident\_df %>% select(-c(accident\_reference, police\_force, local\_authority\_district,

local\_authority\_ons\_district, local\_authority\_highway, first\_road\_number,

second\_road\_number, lsoa\_of\_accident\_location, did\_police\_officer\_attend\_scene\_of\_accident))

names(accident\_df)

#NA variables are indicated as -1 or NULL for some variables- making NA

accident\_df[accident\_df==-1]<- NA

accident\_df[is.null(accident\_df)]<- NA

#visualising missing data

vis\_miss(accident\_df, warn\_large\_data = FALSE)

colSums(is.na(accident\_df))

# remove second\_road\_class and junction\_contol

accident\_df <- accident\_df %>% select(-c(second\_road\_class, junction\_control))

names(accident\_df)

#CONVERT TIME TO JUST HOUR

accident\_df$hour<- gsub("\\:.\*","",accident\_df$time)

#make integer

accident\_df$hour<- as.integer(accident\_df$hour)

#CONVERT DATE TO MONTH

#convert to date

accident\_df$date <- as.Date(accident\_df$date, format = "%d/%m/%Y")

#extract month and put in new variable month

accident\_df$month <- format(accident\_df$date, format = "%m")

#check

accident\_df %>% count(month)

#make integer

accident\_df$month<- as.integer(accident\_df$month)

#make speed limit integer

accident\_df$speed\_limit<- as.integer(accident\_df$speed\_limit)

accident\_df$longitude <- as.numeric(accident\_df$longitude)

accident\_df$latitude <- as.numeric(accident\_df$latitude)

accident\_df$location\_northing\_osgr <- as.numeric(accident\_df$location\_northing\_osgr)

accident\_df$location\_easting\_osgr <- as.numeric(accident\_df$location\_easting\_osgr)

accident\_df$number\_of\_vehicles <- as.numeric(accident\_df$number\_of\_vehicles)

accident\_df$number\_of\_casualties <- as.numeric(accident\_df$number\_of\_casualties)

str(accident\_df)

###CORRELATION PLOT

library(corrplot)

accident\_df.corr <- accident\_df %>%

select(-c(accident\_index, date, time)) %>% na.omit() %>% cor()

corrplot(accident\_df.corr)

heatmap(x = accident\_df.corr, symm = TRUE)

# speed limit and urban/rural area seem correlated

###VISUALISE NUMERIC DATA

accident\_df$accident\_severity <- as.factor(accident\_df$accident\_severity)

histogram <- function(attribute) {

accident\_df %>% ggplot(aes(x= attribute,fill = accident\_severity)) +

geom\_histogram(bins = 30L) +

scale\_fill\_hue(direction = 1) +

theme\_minimal()

}

nos <- histogram(accident\_df$location\_northing\_osgr) + labs(x = "locaton\_northing\_osgr")

eos <- histogram(accident\_df$location\_easting\_osgr) + labs(x = "locaton\_easting\_osgr")

long<- histogram(accident\_df$longitude) + labs(x = "longitude")

lat <- histogram(accident\_df$latitude) + labs(x = "latitude")

hour <- histogram(accident\_df$hour) + labs(x = "hour")

mon <- histogram(accident\_df$month) + labs(x = "month")

cas <- histogram(accident\_df$number\_of\_casualties) + labs(x = "number\_of\_casualties")

veh <- histogram(accident\_df$number\_of\_vehicles) + labs(x = "number\_of\_vehicles")

library(ggpubr)

ggarrange(nos, eos, lat, long,

labels = c("A", "B", "C", "D"),

ncol = 2, nrow = 2)

ggarrange(hour, mon, cas, veh,

labels = c("E", "F","G", "H"),

ncol = 2, nrow = 2)

#remove date and time as created new variables

#remove location\_easting and location\_northing as latitude and longitude similar so can be used in place

accident\_df <- accident\_df %>% select(-c(date,time,location\_easting\_osgr, location\_northing\_osgr))

###CHANGE CATEGORICAL FORMAT

#accident\_severity

#check levels

accident\_df %>% count(accident\_severity)

#check missing data (-1)

sum(is.na(accident\_df$accident\_severity))

#change value

accident\_df$accident\_severity <- as.integer(accident\_df$accident\_severity)

accident\_df$accident\_severity[accident\_df$accident\_severity==1] <- "FATAL"

accident\_df$accident\_severity[accident\_df$accident\_severity==2] <- "SERIOUS"

accident\_df$accident\_severity[accident\_df$accident\_severity==3] <- "SLIGHT"

#convert to factor

accident\_df$accident\_severity <- as.factor(accident\_df$accident\_severity)

#first\_road\_class

#check NA

sum(is.na(accident\_df$first\_road\_class))

#make unkown NA

accident\_df$first\_road\_class[accident\_df$first\_road\_class==6]<-NA

#check factor levels

accident\_df %>% count(first\_road\_class)

#change level name

accident\_df$first\_road\_class[accident\_df$first\_road\_class==1]<-"MOT"

accident\_df$first\_road\_class[accident\_df$first\_road\_class==2]<-"MOT"

accident\_df$first\_road\_class[accident\_df$first\_road\_class==3]<-"A"

accident\_df$first\_road\_class[accident\_df$first\_road\_class==4]<-"B"

accident\_df$first\_road\_class[accident\_df$first\_road\_class==5]<-"C"

#make factor

accident\_df$first\_road\_class <- as.factor(accident\_df$first\_road\_class)

#road\_type

#check NA

sum(is.na(accident\_df$road\_type))

#create NA VARIABLES

accident\_df$road\_type[accident\_df$road\_type==9]<- NA

accident\_df$road\_type[accident\_df$road\_type==-1]<- NA

accident\_df$road\_type[accident\_df$road\_type==12]<- NA

#check factor levels

accident\_df %>% count(road\_type)

#change variable names to make interpretation easier

accident\_df$road\_type[accident\_df$road\_type==1]<-"RB"

accident\_df$road\_type[accident\_df$road\_type==2]<-"OWS"

accident\_df$road\_type[accident\_df$road\_type==3]<-"DC"

accident\_df$road\_type[accident\_df$road\_type==6]<-"SC"

accident\_df$road\_type[accident\_df$road\_type==7]<-"SR"

#make factor

accident\_df$road\_type<-as.factor(accident\_df$road\_type)

#speed limit

accident\_df %>% count(speed\_limit)

#NULL didn't work earlier so make NA using string "NULL"

accident\_df$speed\_limit[accident\_df$speed\_limit=="NULL"] <- NA

#check NA

sum(is.na(accident\_df$speed\_limit))

#convert to factor

accident\_df$speed\_limit<- as.factor(accident\_df$speed\_limit)

#junction\_detail

#check NA

sum(is.na(accident\_df$junction\_detail))

#change to NA

accident\_df$junction\_detail[accident\_df$junction\_detail==99]<-NA

accident\_df$junction\_detail[accident\_df$junction\_detail==9]<-NA

#check factor levels

accident\_df %>% count(junction\_detail)

#change level names to make interpretation easier

accident\_df$junction\_detail[accident\_df$junction\_detail==0]<-"NOT"

accident\_df$junction\_detail[accident\_df$junction\_detail==1]<-"RB"

accident\_df$junction\_detail[accident\_df$junction\_detail==2]<-"MRB"

accident\_df$junction\_detail[accident\_df$junction\_detail==3]<-"TJUN"

accident\_df$junction\_detail[accident\_df$junction\_detail==5]<-"SR"

accident\_df$junction\_detail[accident\_df$junction\_detail==6]<-"CR"

accident\_df$junction\_detail[accident\_df$junction\_detail==7]<-"4ARM"

accident\_df$junction\_detail[accident\_df$junction\_detail==8]<-"PD"

#make factor

accident\_df$junction\_detail<-as.factor(accident\_df$junction\_detail)

#pedestrian\_crossing\_human\_control

#check NA

sum(is.na(accident\_df$pedestrian\_crossing\_control))

#change to NA

accident\_df$pedestrian\_crossing\_human\_control[accident\_df$pedestrian\_crossing\_human\_control==9]<-NA

#change level name - make yes or no due to low levels of not none answers

accident\_df$pedestrian\_crossing\_human\_control[accident\_df$pedestrian\_crossing\_human\_control==0]<- "NO"

accident\_df$pedestrian\_crossing\_human\_control[accident\_df$pedestrian\_crossing\_human\_control==1]<- "YES"

accident\_df$pedestrian\_crossing\_human\_control[accident\_df$pedestrian\_crossing\_human\_control==2]<- "YES"

#make factor

accident\_df$pedestrian\_crossing\_human\_control <- as.factor(accident\_df$pedestrian\_crossing\_human\_control)

#check levels

accident\_df %>% count(pedestrian\_crossing\_human\_control)

#pedestrian physical control

#check NA

sum(is.na(accident\_df$pedestrian\_crossing\_physical\_facilities))

#make unknown NA

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==9]<-NA

#change level name - change to yes or no for same reason as other pedestrian

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==0] <- "NO"

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==1] <- "YES"

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==4] <- "YES"

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==5] <- "YES"

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==7] <- "YES"

accident\_df$pedestrian\_crossing\_physical\_facilities[accident\_df$pedestrian\_crossing\_physical\_facilities==8] <- "YES"

#make factor

accident\_df$pedestrian\_crossing\_physical\_facilities <- as.factor(accident\_df$pedestrian\_crossing\_physical\_facilities)

#check levels

accident\_df %>% count(pedestrian\_crossing\_physical\_facilities)

#light conditions

#check Na

sum(is.na(accident\_df$light\_conditions))

#convert to NA

accident\_df$light\_conditions[accident\_df$light\_conditions==7]<-NA

#check factor levels

accident\_df %>% count(light\_conditions)

#change level name

accident\_df$light\_conditions[accident\_df$light\_conditions==1]<-"DAY"

accident\_df$light\_conditions[accident\_df$light\_conditions==4]<-"LIT"

accident\_df$light\_conditions[accident\_df$light\_conditions==5]<-"UNLIT"

accident\_df$light\_conditions[accident\_df$light\_conditions==6]<-"NL"

#convert to factor

accident\_df$light\_conditions<-as.factor(accident\_df$light\_conditions)

#weather\_conditions

#check NA

sum(is.na(accident\_df$weather\_conditions))

#convert to NA

accident\_df$weather\_conditions[accident\_df$weather\_conditions==9]<-NA

accident\_df$weather\_conditions[accident\_df$weather\_conditions==8]<-NA

#check levels

accident\_df %>% count(weather\_conditions)

#high\_winds

accident\_df$high\_winds<-NA

for(i in 1:nrow(accident\_df)) {

if (is.na(accident\_df$weather\_conditions[i])) {

accident\_df$high\_winds[i] <- NA

}

else if ((accident\_df$weather\_conditions[i]<4) || (accident\_df$weather\_conditions[i]==7)){

accident\_df$high\_winds[i] <- "NO"

}

else if ((accident\_df$weather\_conditions[i]>3) & (accident\_df$weather\_conditions[i]<7)){

accident\_df$high\_winds[i] <- "YES"

}

}

accident\_df %>% count(high\_winds)

#combine high winds and non high-winds

accident\_df$weather\_conditions[accident\_df$weather\_conditions==4]<-1

accident\_df$weather\_conditions[accident\_df$weather\_conditions==5]<-2

accident\_df$weather\_conditions[accident\_df$weather\_conditions==6]<-3

#change level names

accident\_df$weather\_conditions[accident\_df$weather\_conditions==1]<-"FINE"

accident\_df$weather\_conditions[accident\_df$weather\_conditions==2]<-"RAIN"

accident\_df$weather\_conditions[accident\_df$weather\_conditions==3]<-"SNOW"

accident\_df$weather\_conditions[accident\_df$weather\_conditions==7]<-"FOG"

#check levels

accident\_df %>% count(weather\_conditions)

#make factor

accident\_df$weather\_conditions<-as.factor(accident\_df$weather\_conditions)

#road\_surface\_conditions

#CHECK na

sum(is.na(accident\_df$road\_surface\_conditions))

#convert to NA

accident\_df$road\_surface\_conditions[accident\_df$road\_surface\_conditions==9]<-NA

#check levels

accident\_df %>% count(accident\_df$road\_surface\_conditions)

#change level names

accident\_df$road\_surface\_conditions[accident\_df$road\_surface\_conditions==1]<-"DRY"

accident\_df$road\_surface\_conditions[accident\_df$road\_surface\_conditions==2]<-"WET"

accident\_df$road\_surface\_conditions[accident\_df$road\_surface\_conditions==3]<-"SNOW"

accident\_df$road\_surface\_conditions[accident\_df$road\_surface\_conditions==4]<-"ICE"

accident\_df$road\_surface\_conditions[accident\_df$road\_surface\_conditions==5]<-"FLOOD"

#make factor

accident\_df$road\_surface\_conditions<-as.factor(accident\_df$road\_surface\_conditions)

#special conditions

#check NA

sum(is.na(accident\_df$special\_conditions\_at\_site))

#convert NA

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==9]<-NA

#check levels

accident\_df %>% count(special\_conditions\_at\_site)

#change levels name

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==0]<-"NONE"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==1]<-"AFSO"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==2]<-"ASPD"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==3]<-"RSO"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==4]<-"RW"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==5]<-"RSD"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==6]<-"OIL"

accident\_df$special\_conditions\_at\_site[accident\_df$special\_conditions\_at\_site==7]<-"MUD"

#make factor

accident\_df$special\_conditions\_at\_site<-as.factor(accident\_df$special\_conditions\_at\_site)

#carriageway hazards

#check NA

sum(is.na(accident\_df$carriageway\_hazards))

#convert NA

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==9]<-NA

#check levels

accident\_df %>% count(carriageway\_hazards)

#change level names

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==0]<-"NONE"

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==1]<-"LOAD"

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==2]<-"OBJ"

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==3]<-"ACC"

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==6]<-"PED"

accident\_df$carriageway\_hazards[accident\_df$carriageway\_hazards==7]<-"ANI"

#convert to factor

accident\_df$carriageway\_hazards<-as.factor(accident\_df$carriageway\_hazards)

#urban or rural

#check NA

sum(is.na(accident\_df$urban\_or\_rural\_area))

#check levels

accident\_df %>% count(urban\_or\_rural\_area)

#make unallocated NA

accident\_df$urban\_or\_rural\_area[accident\_df$urban\_or\_rural\_area==3]<-NA

#change level name

accident\_df$urban\_or\_rural\_area[accident\_df$urban\_or\_rural\_area==1]<-"URB"

accident\_df$urban\_or\_rural\_area[accident\_df$urban\_or\_rural\_area==2]<-"RUR"

#make factor

accident\_df$urban\_or\_rural\_area<-as.factor(accident\_df$urban\_or\_rural\_area)

#trunk\_road\_flag

#check NA

sum(is.na(accident\_df$trunk\_road\_flag))

#change level name

accident\_df$trunk\_road\_flag[accident\_df$trunk\_road\_flag==1] <- "TRUNK"

accident\_df$trunk\_road\_flag[accident\_df$trunk\_road\_flag==2] <- "NOT"

#make factor

accident\_df$trunk\_road\_flag <- as.factor(accident\_df$trunk\_road\_flag)

#check levels

accident\_df %>% count(trunk\_road\_flag)

##VISUALISE ACCIDENT SEVERITY DIFFERENCES IN NUMERICAL DATA

#histograms

accident\_df$longitude <- as.numeric(accident\_df$longitude)

#VISUALISE ACCIDENT SEVERITY DIFFERENCES IN CATEGORICAL DATA

#(modified from julia silge's blog)

str(accident\_df)

accident\_df <- accident\_df %>% mutate\_if(is.integer,as.factor)

accident\_df <- accident\_df %>% mutate\_if(is.numeric,as.factor)

accident\_df %>%

select(accident\_severity, accident\_year,day\_of\_week,first\_road\_class,

road\_type,speed\_limit,junction\_detail,

) %>% pivot\_longer(accident\_year:junction\_detail) %>%

ggplot(aes(y = value, fill = accident\_severity)) +

geom\_bar(position = "fill") +

facet\_wrap(vars(name), scales = "free", ncol = 2) +

labs(x = NULL, y = NULL, fill = NULL)

accident\_df %>%

select(accident\_severity, pedestrian\_crossing\_human\_control, pedestrian\_crossing\_physical\_facilities,

light\_conditions,weather\_conditions

) %>% pivot\_longer(pedestrian\_crossing\_human\_control:weather\_conditions) %>%

ggplot(aes(y = value, fill = accident\_severity)) +

geom\_bar(position = "fill") +

facet\_wrap(vars(name), scales = "free", ncol = 2) +

labs(x = NULL, y = NULL, fill = NULL)

accident\_df %>%

select(accident\_severity, high\_winds, road\_surface\_conditions,

urban\_or\_rural\_area,trunk\_road\_flag

) %>% pivot\_longer(high\_winds:trunk\_road\_flag) %>%

ggplot(aes(y = value, fill = accident\_severity)) +

geom\_bar(position = "fill") +

facet\_wrap(vars(name), scales = "free", ncol = 2) +

labs(x = NULL, y = NULL, fill = NULL)

accident\_df %>%

select(accident\_severity,carriageway\_hazards, special\_conditions\_at\_site

) %>% pivot\_longer(carriageway\_hazards:special\_conditions\_at\_site) %>%

ggplot(aes(y = value, fill = accident\_severity)) +

geom\_bar(position = "fill") +

facet\_wrap(vars(name), scales = "free", ncol = 2) +

labs(x = NULL, y = NULL, fill = NULL)

#change day of week to whether week day or weekend

#check NA

sum(is.na(accident\_df$day\_of\_week))

#change to weekday or not

accident\_df$day\_of\_week <- as.integer(accident\_df$day\_of\_week)

accident\_df$day\_of\_week[accident\_df$day\_of\_week==1] <- "NO"

accident\_df$day\_of\_week[accident\_df$day\_of\_week==7] <- "NO"

accident\_df$day\_of\_week[accident\_df$day\_of\_week==2] <- "YES"

accident\_df$day\_of\_week[accident\_df$day\_of\_week==3] <- "YES"

accident\_df$day\_of\_week[accident\_df$day\_of\_week==4] <- "YES"

accident\_df$day\_of\_week[accident\_df$day\_of\_week==5] <- "YES"

accident\_df$day\_of\_week[accident\_df$day\_of\_week==6] <- "YES"

#make factor

accident\_df$day\_of\_week <- as.factor(accident\_df$day\_of\_week)

#check factor levels

accident\_df %>% count(day\_of\_week)

#CHECK MISSING VALUES AGAIN

vis\_miss(accident\_df, warn\_large\_data = FALSE)

colSums(is.na(accident\_df))

#remove road\_class\_type

prep\_data <- accident\_df %>% select(-c(first\_road\_class))

prep\_data <- na.omit(prep\_data)

write.csv(prep\_data, "prepped\_data.csv",row.names = FALSE)

**Appendix B**

R code for decision tree and random forest implementation.

#CLASSIFICATION

library(tidyverse)

library(rsample)

library(caret)

library(party)

library(randomForest)

rm(list=ls())

#set working directory

setwd("C:\\Users\\smann\\OneDrive\\University\\Data Science\\ASDM\\Project\\Road safety")

#import datasets

accident\_df <- read.csv("prepped\_data.csv")

str(accident\_df)

###REMOVE accident\_index, accident\_year, longitude, latitude

accident\_df <- accident\_df %>% select(-c(accident\_index, accident\_year,

longitude, latitude))

#make all factors

str(accident\_df)

accident\_df <- accident\_df %>% mutate\_if(is.character,as.factor)

accident\_df <- accident\_df %>% mutate\_if(is.integer,as.factor)

accident\_df$number\_of\_vehicles <- as.integer(accident\_df$number\_of\_vehicles)

accident\_df$number\_of\_casualties <- as.integer(accident\_df$number\_of\_casualties)

accident\_df$month <- as.integer(accident\_df$month)

accident\_df$hour <- as.integer(accident\_df$hour)

#split dataset

set.seed(123)

#create testing and training sets

accident\_split <- accident\_df %>% initial\_split(prop=0.7, strata=accident\_severity)

accident\_train <- training(accident\_split)

accident\_train <- downSample(accident\_train, accident\_train$accident\_severity)

accident\_train <- accident\_train %>% select(-Class)

accident\_test <- testing(accident\_split)

accident\_test %>% count(accident\_severity)

accident\_train %>% count(accident\_severity)

##DECISION TREE

control <- trainControl(method = "cv", number = 10)

###use cross validation to find optimal tree

accident\_dt <- train(accident\_severity~., accident\_train, method="ctree",

trControl= control)

accident\_dt

###employ decision tree using suggested mincriterion

accidentTree <- ctree(accident\_severity~.,data=accident\_train,

control = ctree\_control(mincriterion = 0.5,

maxdepth = 5))

print(accidentTree)

plot(accidentTree)

#get model accuracy

accidentPred <- predict(accidentTree, newdata= accident\_test)

confusionMatrix(accidentPred, accident\_test$accident\_severity, mode = "everything")

##RANDOM FOREST

##random forest with cross validation

accident\_rf <- train(accident\_severity~., accident\_train, method="rf",

trControl= control)

accident\_rf

#random forest optimal model

accidentRF <- randomForest(accident\_severity~.,data=accident\_train,

mtry=2)

accidentRF

plot(accidentRF)

varImpPlot(accidentRF)

importance(accidentRF)

#accuracy

accidentPred <- predict(accidentRF, newdata= accident\_test)

confusionMatrix(accidentPred, accident\_test$accident\_severity, mode = "everything")