ASSESSMENT 2: DATA ANALYSIS

Part B: Report

36103 Statistical Thinking for Data Science

University Of Technology Sydney

Group Name

Retail Group

Group Members

Leah Nguyen

Tony Tan

Anmol Mittal

Ben McKinnon

Kasun Caldera

Manasa Burli Nagendra

Date

October, 2021

Contents

1 Executive Summary

1.1 Overview

Modeling the severity of accidents based on the most effective variables accounts for developing a high-precision model presenting the possibility of occurrence of each category of future accidents, and it could be utilized to prioritize the corrective measures for authorities.

The purpose of this study is to identify the variables contributing to the fatal accidents in Victoria, Australia by collecting information on road accidents from 2006 to 2020. In this regard, the multiple logistic regression is used to recognize the most influential variables on the fatal accidents and the approach for accident prediction.

1.2 Methodology

Then it was prepared to handle its outliers, missing values and skewness. This was followed with further transformation of the data using one hot encoding and label encoding. And then we fixed the data imbalance by using two different techniques, undersampling and SMOTE, to create two different datasets. Several iterations of these three steps were taken to conduct this regression analysis:

- **Step 1:** Select features that go into the model
- **Step 2:** Develop and adjust model
- **Step 3:** Assess and validate model

Our final model uses the undersampling data set. Training set is from the years 2006 to 2017, and the testing set is from the years 2017 to 2020. With a prediction threshold set as 0.50.

1.3 Overview of the result

Results show that the logistic regression in the forward stepwise method has an accuracy prediction power of 72.4%. The most important result of the logit model accentuates the role of variables for speed zone, vulnerability of pedestrians, number of people involved (more distractions), alongside unfavorable weather and the dominant role of unsafe and poor quality of vehicles on increasing the severity of accidents.

1.4 Limitations

Our limitations of this work were the limited availability of complete data on which to conduct the analysis. So, while the analysis produced non-significance, it is anticipated that as more data becomes available, the models will yield more concrete findings.

Regardless, understanding the relationships among incident causal factors and outcomes may shed light on those causal factors which have the potential to lead to catastrophic events and those which may lead to less severe events.

2 Introduction

2.1 Global Road Fatalities

Road traffic accidents are the leading cause of injuries and deaths worldwide. According to the WHO, road traffic accidents kill roughly 1.3 million people and injure up to 50 million people each year (World Health Organization, 2021).

If current trends continue, road traffic accidents are expected to be the third leading cause of disease and injury worldwide by 2020. (Murray el al., 1996). The burden of traffic-related fatalities, disabilities, and injuries has a significant impact on the health and social and economic development of many countries, particularly low and middle-income countries (Nantulya et al., 2002).

2.2 Victoria Road Fatalities

For many decades, Victoria has been a leader in road safety in Australia. Victoria's fatality rate per 100,000 population in December 2019 was 4.06, compared to the national average of 4.68. (Australian Automobile Association, 2020). However, this does not change the fact that progress in decreasing the road toll in Victoria has halted, and motor vehicle crashes remain a major cause of death and injury, with significant health and cost consequences.

Having said that, the entire annual cost of road casualties in Victoria is projected to be well over $1 billion (including property damage), and the total lifetime medical and associated costs for motor vehicle traffic-related injury in 1993-94 were $570 million (VicRoads and TAC, 1997; Watson and Ozanne- Smith; 1997). Furthermore, Minister for Road Safety – Luke Donnellan (Towards Zero, 2020) indicated that over the next ten years, roughly 2,500 people in Victoria will die in automobile accidents, and 50,000 people will be hospitalized with serious and life-changing injuries.

Despite the fact that the Victoria management authority has been addressing the situation by enacting a number of rules and safety measures, serious road accidents are nearly always the result of driver carelessness and irresponsibility. This requires an analysis of the relative influence of factors on the number of deaths in Victoria in order to develop effective methods to minimize the rate of road fatalities.

Therefore, this study aims to:

- Analyze crash fatalities data in Australia according to several factors, including age, causes of crash deaths, and type of road users, locations, etc. to finding factors related to fatalities conditioned on there being an accident in the first place.

- Develop a prediction model for crash fatalities in Australia based on crash fatalities data for the period 2006-2020 by using the logistics modelling method.

3 Data

3.1 Accident Data

The national road crash database obtained from the Road Crash Information System was the major source of data used in this study (RCIS). The database contains 9 tables, 500,000+ crash records, and 100 features spanning a 4-year period from 2006 to 2020. (Appendix A-1, A-2).

With such a vast number of features, we can investigate several methods for predicting the fatal probability associated with different factors. The dataset is made up of features in multiple formats that include both numerical and categorical data. Because the aim of this research project was to discover the factors that potentially contribute to fatal road injuries, data was refined and parsed in the Data Pre-processing section for further analysis.

3.2 Weather Data

In order to assist our analysis, our group has also gathered weather data from Victoria from 2006 to 2020. In our study, we adopted the Visual Crossing Weather API to make an API request for the location based on a collection of all the distinct postcodes from the node dataset. Due to the time required to scrape data from the API, the program has divided down the collection to collect weather data for all suburbs yearly. The collection contains approximately 1,000,000+ weather records in various forms, each with 25 features (Appendix A-3).

As a result, the weather data for each year was combined, the address columns were split to match by postcode, and the postcode was eventually matched. This will also be included in the merge data set.

3.3 Overview of final dataset

Since the API response contains a list of records for the date range specified in the query, we then use the location's postcode and the date to match each of our accident data to the weather. The final data is a merged version of the accident and weather data with the features of interest that were chosen. It is made up of 52 features (Appendix A-4) and 550,578 records, with each record representing a victim involved in an accident.

4 Exploratory Data Analysis

To facilitate analysis and discovery of insights from the road accident dataset, our team classifies the findings into three major areas of questions:

- The “When” - Period of Time
- The “Where” - Geographical Location
- The “How” - Additional Factors

Additionally, we will filter the datasheet to include only values with “SEATING POSITION” equal to drivers, as this will aid us in determining the cause of road deaths much better.

4.1 The “When” - Time

From 2006 until 2020, the number of deaths is depicted in Figure 1, with each month represented separately. The months of February to May are the most deadly. By contrast, during the year, July and September are the months of have the fewest fatalities.

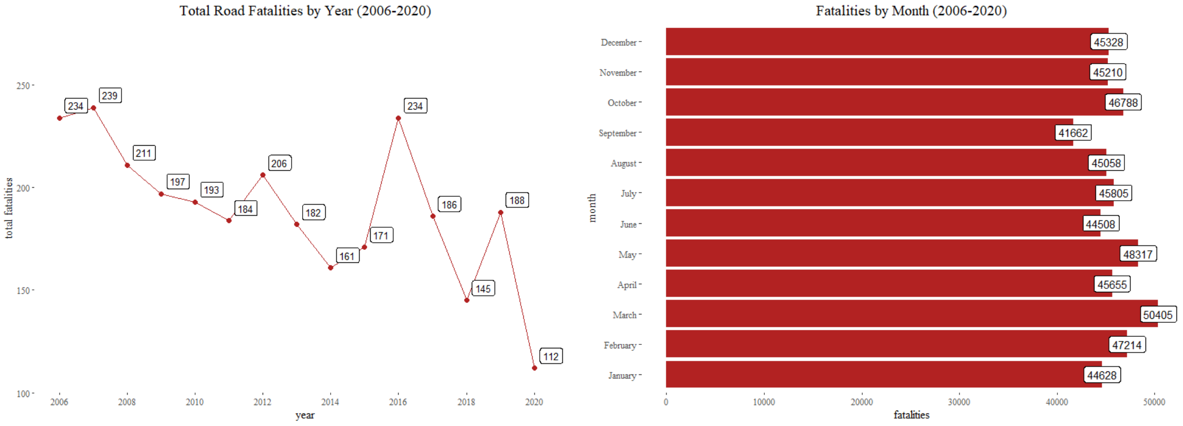

Figure 1: Fatalities by Years and Months

The intensity of fatalities at each hour of the day is depicted in the heat map to the right (Figure 2). On most days, the biggest number of deaths occurs at 15:00, when the most people are at their most vulnerable. However, it is possible that this is due to the school’s dismissal time, during which the school zone measures are enforced.

In terms of fatalities by time, it shown that the fatalities are high during the weekdays at 15:00 and 16:00. However, during the weekends, the trend starts increasing from 12:00 to 15:00 and declines from there.

The trend in fatalities throughout the weekdays and weekends is depicted by the candlestick charts shown above. The number of fatalities is highest during the weekday afternoons and evenings between 15:00 and 16:00. However, on weekends, the trend begins to increase from 12:00 to 15:00 and then begins to decline from there.

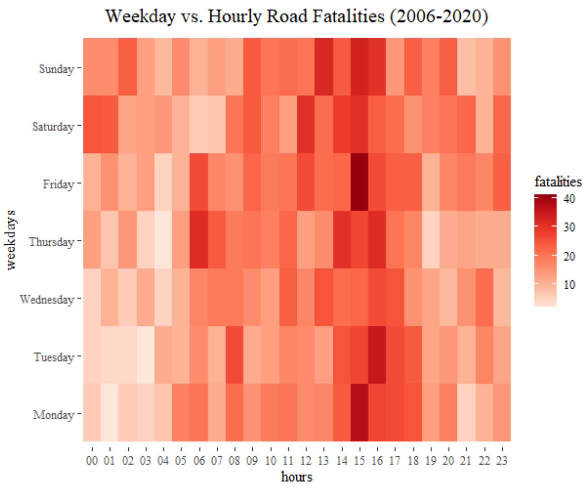

Figure 2: Weekday vs Hourly Fatalities (2006-2020)

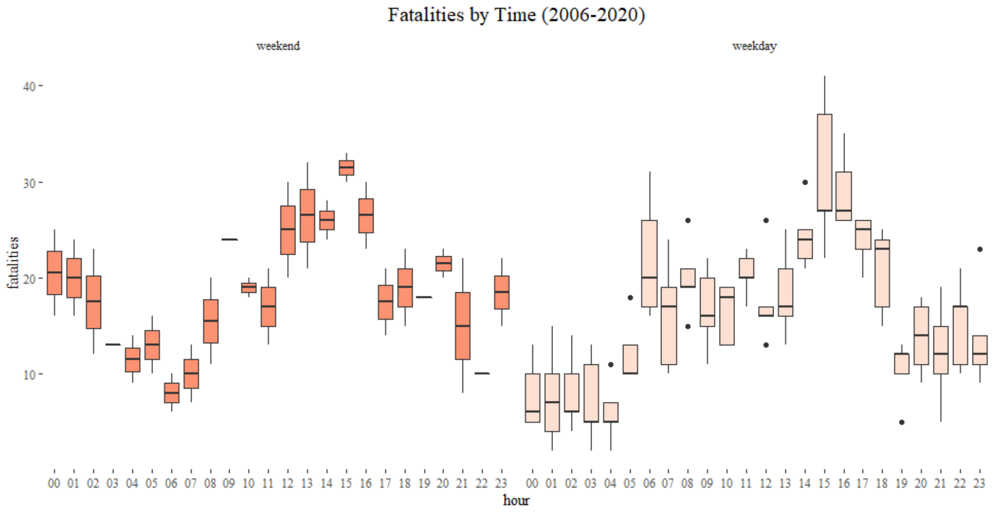

Figure 3: Fatalities by Time (2006-2020)

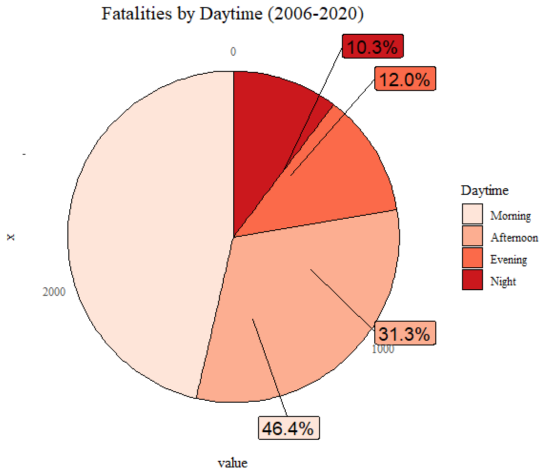

Figure 4: Fatalities by Daytime (2006-2020)

4.2 The "Where" – Location

Figure 5 demonstrates the top 10 LGA locations with the highest number of deaths in Victoria. "GEELONG", "CASEY" and "YARRA RANGES" appear to be the top 3 LGA areas with the highest fatalities number from road accidents. However, there is no direct connections between the high number of road toll with the level of road safety in these areas since the data could be biased toward the population density since the more people living there will possibly have high chances of getting higher number of road accidents.

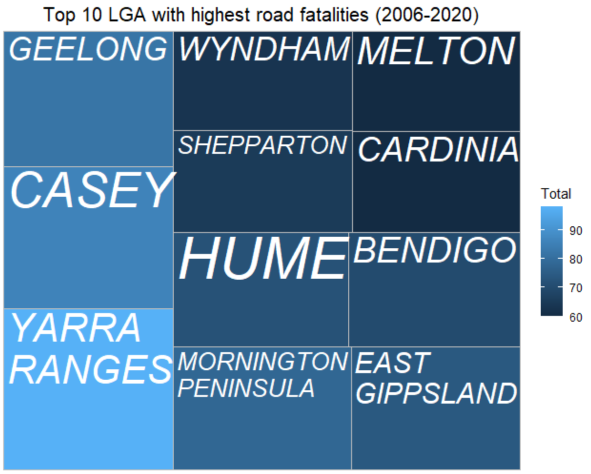

Figure 5: Top 10 LGA areas with highest road toll (2006-2020)

Additionally, in terms of casual relationships between location and fatality rate, another thing should be noticed is that the vast majority of fatalities have occurred on routes that do not have any intersections. As a result, the majority of accidents occur on highways. In addition, there are numerous fatalities at the 'T' and 'Cross' intersections.

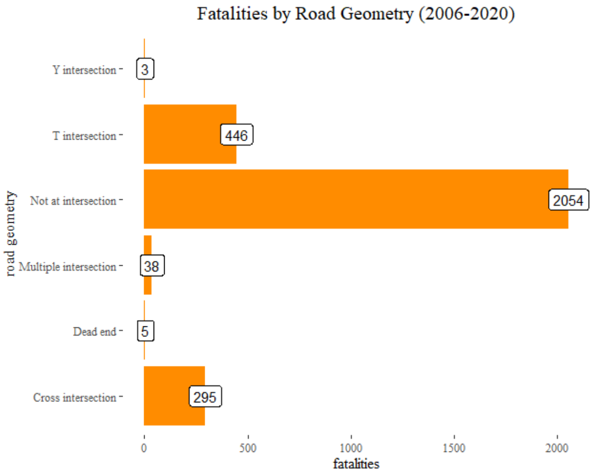

Figure 6: Fatalities by Road Geometry (2006-2020)

4.3 The “Why” – Other factors

It is pretty obvious that the majority of fatalities happened on highways with speed limits of 100km/h or above, which makes sense given that a motorist is unlikely to survive a crash at such a high rate of speed (Figure 7). The second biggest number of fatalities occurs on highways with speed restrictions of 80km/h and 60km/h, respectively.

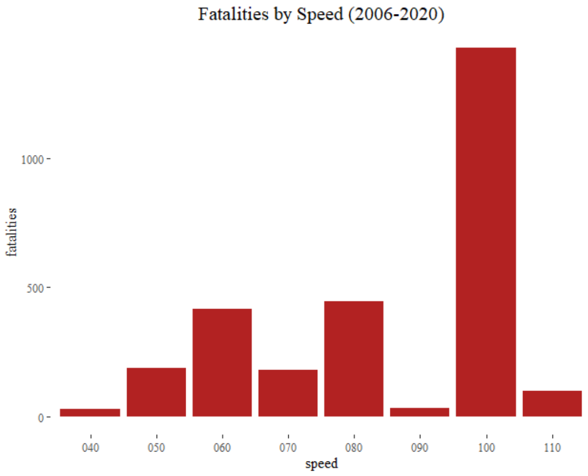

Figure 7: Fatalities by Speed Limits (2006-2020)

In terms of death by road users, motor vehicle drivers accounted for the vast majority of fatalities (73 percent), followed by motorcycle riders (22 percent) and bicycles (14 percent) (5 percent). A further concern is that nearly half of the fatalities were caused by a collision with another motor vehicle. In the second largest number of fatalities, a collision with a stationary object on the road was the cause.

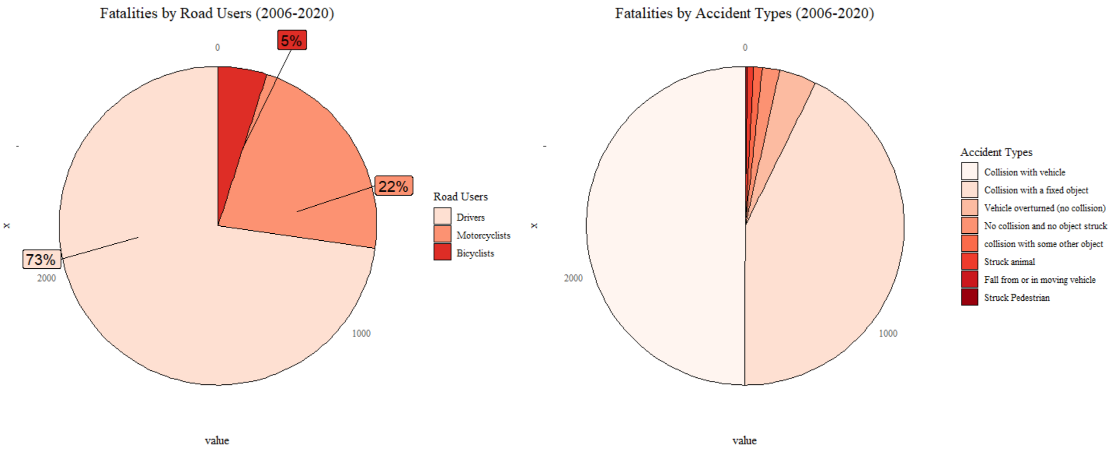

Figure 8: Fatalities by Road Users and Accident Types (2006-2020)

Considering road surface conditions, muddy roads were the most dangerous, followed by slippery and wet roads, which accounted for the majority of fatalities. However, it is said that there is no strong differences of the number of road tolls between weather conditions even though Rain and Cloudy appear to have more car fatalities as the number is not very significant.

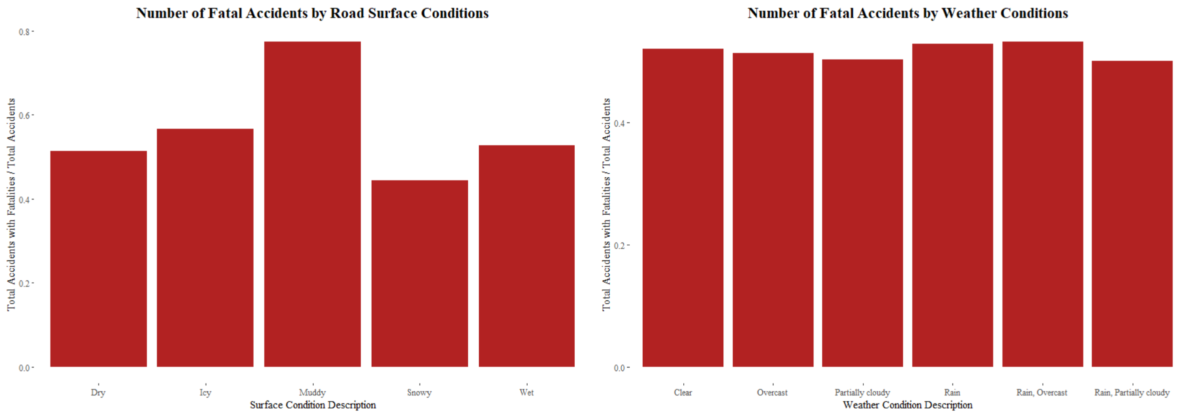

Figure 9: Fatalities by Surface Conditions and Weather Conditions (2006-2020)

5 Method

5.1 Regression Model

In our study, we chose Logistic Regression Model to tackle the research questions. According to Menard (2002), logistic regression models are used to model the probability of a certain event based on independent predictor variables.

While more sophisticated machine learning (ML) techniques have arisen and been applied to accident investigations over the last decade, logistic regression offers several advantages over ML techniques that support its use in our group study.

To begin, regression model findings are easy to interpret. Only the independent variables and their coefficients are required to represent the model in a single formula. The model's coefficients can be used to determine critical variables, as well as the amount and direction of association between each independent variable (i.e., risk factor) and the dependent variable (fatalities rate).

Second, once risk factors are discovered, creating a logistic regression model is straightforward and does not require tweaking multiple hyperparameters, as machine learning methods do. Due to this property, logistic regression is frequently used as the first classifier in predictive research and serves as a valid baseline for more advanced classifiers.

Third, while association rules might be effective for identifying latent patterns in huge data sets, they are fundamentally distinct from classification methods such as logistic regression modeling.

5.2 Data Preparation

5.2.1 Data Filtering

Traditionally, building statistical models starts with selecting variables that can result in a parsimonious model (i.e., having as few variables as possible). In order to do that, the first step is to edit it so that each point is genuinely useful, as larger is not necessarily better (Christensen, 2020). One simple solution is to clearly understand what problems we expect to resolve from our dataset.

Since the aim of this project was to predict the factors that are likely to contribute to deaths from road accidents, we filtered the data points containing the variables 'Drivers' and 'Motorcyclists' of the Road User Types to avoid any bias during analysis. After filtering out, the number of rows is reduced to 311,199 records.

5.2.2 Data Pre-processing

Next, an essential component of statistical modelling is analyzing the data set to ensure the data is tidy and in a compliant format for desired modelling technique. Our merged dataset is unstructured and contains significant superfluous data (defined as not contributing significantly to the prediction process). Since large datasets require longer training times, data preprocessing is, therefore, required to overcome this limitation. Preprocessing involves various tasks including dealing with following problems:

- Handling Outliers
- Handling Missing Values
- Handling Skewness
- Encoding
- Data Imbalance

Therefore, within the dataset, the following data issues were identified and handled accordingly.

Dealing with Missing Values

Handling missing values within the data can be tedious. While some methods can be convenient, they can also be at the expense of the data's integrity. For instance, while handling numerical variables was straightforward, dealing with missing categorical data had challenges.

In our dataset, there are total of 135,303 missing values, which equivalent to 0.83% of our dataset. As illustrated in Figure 10, the majority of the important variables have little or no missing values while most of the missing values are associated with various vehicle information.

In order to deal with missing data, our group has come up with 6 common techniques (Appendix A-5), including:

- Drop / Remove all missing values
- Imputation Using Mode Values
- Imputation Using (Mean/Median) Values

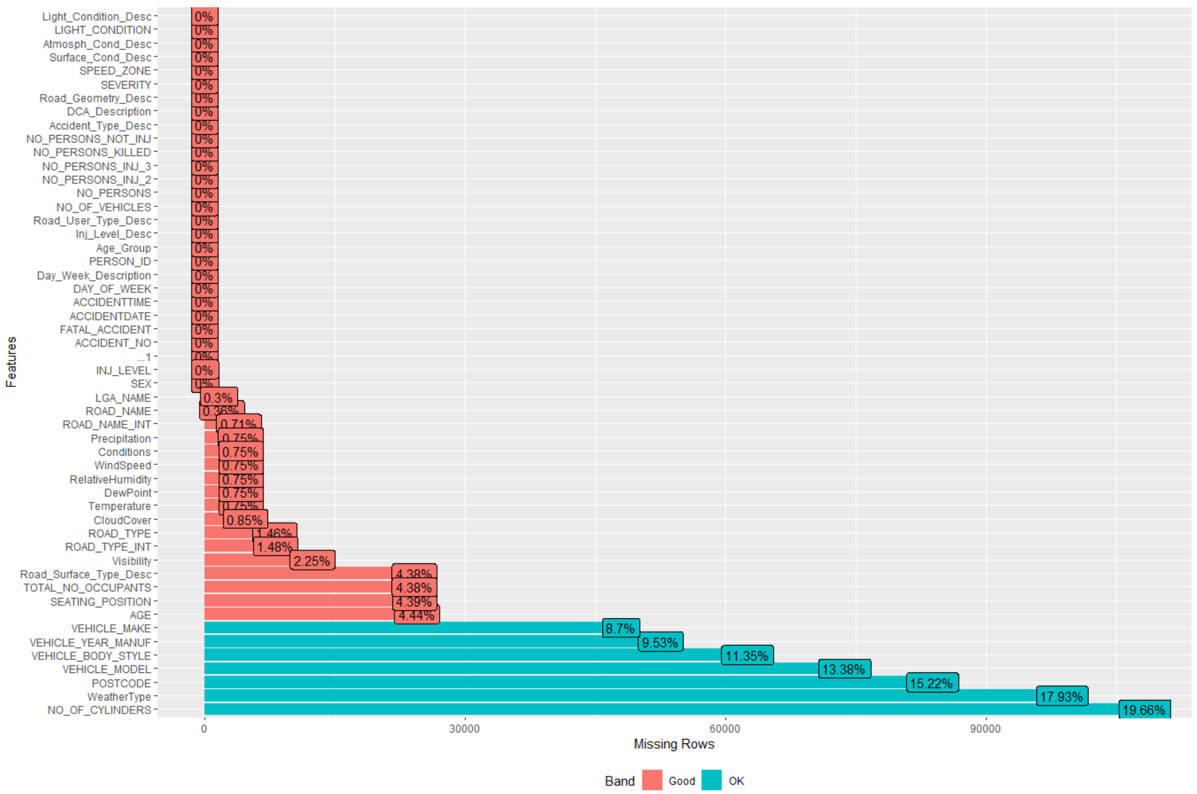

Figure 10: Missing values of the dataset

- Amelia predictive model (Multiple Imputation)
- K-nearest neighbor
- Random Imputation

However, each method has its pros and cons. Based on our research problem and technical requirement, we have chosen to reject some of the approaches as illustrated in Table 1.

Techniques	Problems
Drop / Remove all missing values	"Upon first inspection, multiple variables contained large amounts of missing fields within numerical and categorical data fields. Dropping would result in thousands of lost rows; hence this method was rejected."
Imputation Using Mode Values	"Due to the skewness of several categorical variables, replacing missing values with the mode made this skew even larger; hence this method was rejected."
Imputation Using (Mean/Median) Values	Significantly reduce the model's accuracy and bias the results since it can has an impact on attributes variability.

Techniques	Problems
Amelia predictive model (Multiple Imputation)	“A multiple imputation method that replaces missing values with a bootstrap approach. This approach required 50 gigabytes of system memory to perform, which was resource-intensive for a home computer.”
K-nearest neighbor	“This imputation method was trialed and, similarly to Amelia, is computationally expensive since KNN only works by storing the whole training dataset in memory.”

Table 1. Missing data dealing techniques rejection

As final, we chose **Random Imputation** as the technique for dealing with missing values before constructing regression model (Figure 11). This method eliminates the imputation variance of the estimator of a mean or total, and at the same time preserves the distribution of item values (Chen, et al., 2000).

Comprehensively, this is achieved by using a vector of the cumulative sum of the number of each unique value for that categorical feature and a set of unified random values from 0 to the maximum cumulative sum. And with the set of random numbers, we are then able to impute each missing value according to the index of the cumulative sum list, which effectively is preserving the distribution of the existing values.

Dealing with skewed data

Skewed data is troublesome as logistic regression assumes a normal distribution (Bill, 2014). As illustrated in Figure C-2, there were degrees of skew on some of our numerical data, specifically in precipitation, cloud cover and relative humidity columns. We need to handle this skewness, as our modelling algorithms assumes a normal distribution (Bill, 2014). As our skewed data follows closely to beta distributions, it was appropriate to use log transformation (Hammouri, Sabo and Alsaadawi, 2020) (Appendix C-2).

Standardizing numerical data

Machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique require data to be scaled (Bhandari, 2021). This is important to give each data point greater meaning by transforming the data to comparable scales.

Since our numerical data is spread over on a relatively large scale and needs to be dealt for more effective modelling, standardization need to be employed. In our case, Normal Standardization is the approach that is considered most effective as our numerical data is transformed or already follows the Gaussian Distribution (Lakshmanan, 2019).

As we are rescaling our data for standardization, we use the scale method in R with the default settings, so that the data has a mean of 0 and standard deviation of 1 (Geller, 2019; see also Saporta,

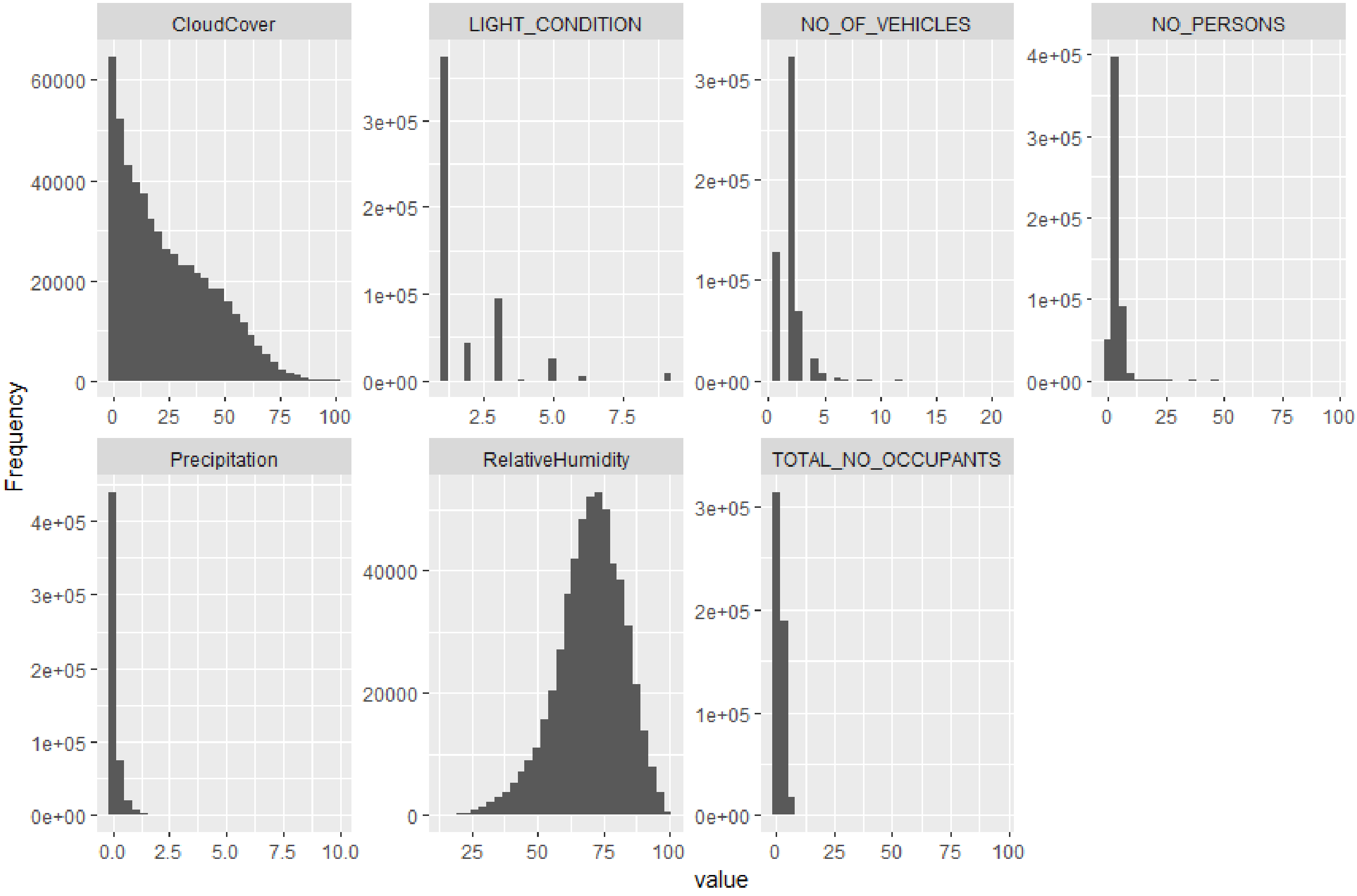

Figure 11: Data distribution of numeric columns

2013; Scale Function - RDocumentation, n.d.).

Handling Outliers

Outliers in data can have significant impacts on analysis of the data. If outliers are present in the data during analysis, they may increase the error variance, reduce the strength of statistical tests, and can bias the results used in estimates of model parameters. Consequently, it is vital to scan and address outliers in the pre-processing phase (Dolgun, 2020).

The decision to remove outliers was based on the belief that they were present due to reporting errors or the presumption that they would impact results. Within our meaningful variables, we found “AGE” column to have several outliers (Figure 12).

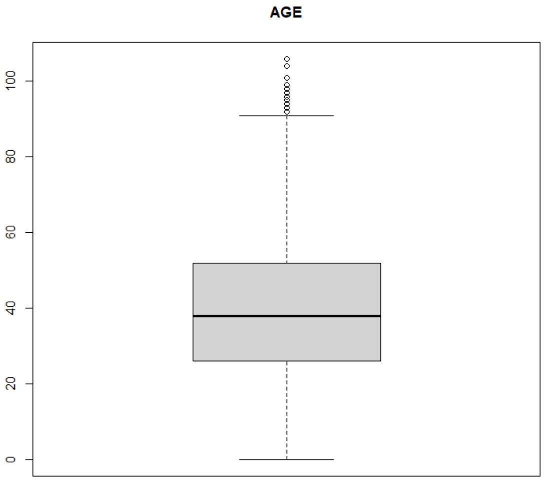

Figure 12: Age Boxplot

Taking a closer group in terms of different age groups, according to Figure 13, all outliers are found in age group 70+ using the univariate box plot approach to detect outliers in AGE for a given “AGE GROUP”. Another strategy is using z-score approach to find out outliers of AGE variable.

First, our group calculated z-scores. Then we find all observations that have the absolute value of AGE’s z-score is greater than 3 (Appendix C-4). From our observation, the minimum z score is -2.4227 and the maximum is 3.9373; there are only 373 out of 311,199 observations, which is only 0.1% of observations, are outliers. In this case, our group decided not to do anything to change the outliers because the age of people should not be changed.

Categorical Data Encoding

In this study, the features collected are the combinations of categorical and numerical data. Categorical variables possess a vast detail of information that links to the target variables; However, as we will use machine learning approach is logistic regression, it cannot operate on categorical values directly and require the input variables and the output variables to be numeric. Therefore, in order to

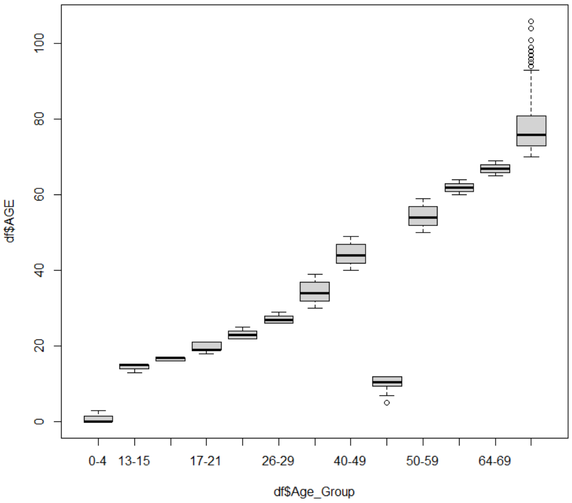

Figure 13: Age Group Multivariate Boxplot

represent categorical information, 2 common techniques as One-Hot Encoding and Label Encoding are proposed (Figure A-6).

Since our categories values contain different variables with little relationships to each other, we adopted this technique to deal with categorical variables, specifically, variables that were deemed to be primarily present in fatal accidents.

Data imbalance

In our dataset, the class imbalance problem presents an important challenge. In particular, it was observed that value counts of non-fatalities and fatalities were 540,266 and 10,312, respectively. In this case, the imbalance data happens to be the “Rare Class Problem”, in which the number of examples of one class is more than the others (Maheshwari et al., 2018).

Working with such imbalanced datasets presents the difficulty that most machine learning algorithms ignore, and so perform poorly on, the minority class, despite the fact that performance on the minority class is often the most significant. Therefore, the chosen methods to solve this problem were:

- **Under sampling:** The class of focus total occurrences were found, and then a random sample of the overpopulated class was taken to create a subsetted data set. This essentially achieves the same as randomly eliminating cases from the majority class (Pykes, 2020).

- **SMOTE:** Oversampling is another way to solve this problem, which has an advantage of no data loss compared to undersampling, however it risks overfitting due to replicated observations (Vidhya, 2020). The article follows up with a better approach called the synthetic minority oversampling technique (SMOTE), which apparently generates artificial data by using bootstrapping and k-nearest neighbors. We can see the introduction and a more in-depth explanation of this technique from Chawla et al. (2002) where page 329 has pseudocode of the logic. Luckily, the article from Vidhya (2020) introduces the DmwR package, which has the SMOTE function to help with applying the SMOTE technique quickly.

5.2.3 Features Selection

We selected our initial features based on patterns of correlations for our target variable observed in EDA. Given, many of these were confirmed based on assumptions made with domain knowledge, since the topic of road traffic accidents is rather universal. To avoid multicollinearity issues, an analysis of the correlation of the selected are performed, where only one of the correlated features were kept.

To reduce computational cost and improve performance of the model, only certain features are kept. For the same reason, this process is performed before applying the techniques to fix data imbalance. The features selected are as follows:

- **Target variable:** FATAL_ACCIDENT
- **User type, which is filtered to be only drivers of cars or motorcyclists:** Road_User_Type_Desc
- **Date and time:** ACCIDENTDATE, ACCIDENTTIME

Features that were produced from one hot encoding (Table 3):

Data Type	Columns	Description	Measure levels
Categorical	Gender	Gender	Female - Male
Categorical	Accident_Type_Desc	Accident Type	Collision.with.a.fixed.object - . . . Struck.animal - . . . Struck.Pedestrian - . . .Vehicle.overturned&no.collision
Categorical	Road_Surface_Type_Desc - Surface_Cond_Desc	Road surface type and condition	Unpaved - Dry - . . . Icy - . . . Muddy - . . . Snowy - . . .Wet
Categorical	Light_Condition_Desc	Light conditions	Dark.No.street.lights - . . . Street.lights.off -

Data Type	Columns	Description	Measure levels
Categorical	Atmosph_Cond_Desc	Weather conditions	Clear - . . . Fog - . . . Raining - . . . Smoke - . . . Strong.winds -
Categorical	Conditions	Conditions	Overcast - . . . Rain - . . . Rain&Overcast
Categorical	Age_Group	Age group	16.17 - . . . 17.21 - . . . 70
Categorical	Day_Week_Description	Weekend	Saturday - . . . Sunday
Numeric	NO_OF_CYLINDERS	Cylinders of car	4 - . . . 6 - . . . 8 - . . . 12
Numeric	NO_OF_VEHICLES	Standardized variables from traffic accidents data	NA
Numeric	NO_PERSONS	Standardized variables from traffic accidents data	NA
Numeric	SPEED_ZONE	Standardized variables from traffic accidents data	NA
Numeric	LIGHT_CONDITION	Standardized variables from traffic accidents data	NA
Numeric	TOTAL_NO_OCCUPANTS	Standardized variables from traffic accidents data	NA
Numeric	CloudCover	Standardized variables from weather data	NA
Numeric	RelativeHumidity	Standardized variables from weather data	NA
Numeric	Precipiation	Standardized variables from weather data	NA
Numeric	EHICLE_YEAR_MANUF	Original variables from traffic accidents data	NA
Numeric	WindSpeed	Original variables from weather data	NA
Numeric	Temperature	Original variables from weather data	NA
Numeric	DewPoint	Original variables from weather data	NA

Table 3. Numeric and categorical features selection for final model

5.3 Developing Model

From the insights we gained from EDA and rationale explained in the Logistic Regression section, we move forward by implementing this regression model to investigate on the factors contributing to the fatal accidents in Victoria.

Firstly, we begin by splitting the training data into train, test and validation sets where training dataset is the sample of data used to fit the model; validation dataset is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters and test dataset is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset (Agrawal, 2021).

The data split ratios for train, test, and validation data are 70/30/10 where training set contains value from 2006-2017, testing set starts from 2017 to 2020 and the cross-validation set is 2020 onwards. However, because our dataset also comes with the rare class problem, the trained data for the logistics model is generated using different methods for dealing with this issue, notably the Under sampling and SMOTE approaches, which was discussed in the Imbalance Data section above. Finally, we will have a total of five datasets to incorporate into our model:

Category	Dataset	Time
Train	No_change_train	2006-2017
	Smote_train	2006-2017
	Under_sample_train	2006-2017
Test	Test_set_generic	2017-2020
Cross Validation	Cross_set_generic	2020 afterwards

Table 4. Datasets for building the model

Furthermore, in order to improve the model's accuracy when evaluating it later, our team considers using the Lift and Reduce method, in which we essentially set the possibility that the model can predict the probability of occurrence for our target variable, which is "FATAL ACCIDENT,' with threshold ratios of 0.40, 0.50, and 0.60, respectively. The model refining process was automated by using the forward selection method. The"stepAIC" function from the "MASS" package was used to achieve this (Appendix D).

6 Evaluation

In the binary-class problem, the confusion matrix (Table 5) displays the results of correctly and incorrectly perceived cases of each class. The confusion matrix is used to assess the classification performance of both common and uncommon classes.

		Predictive class	
		Faulty	**Not Faulty**
Actual class	**Faulty**	True Positive	False Negative
	Not Faulty	False Positive	True Negative

Table 5. Datasets for building the model

In order to evaluate our model's standard evaluation principals are used. The key methods of evaluation being used are:

- **Accuracy:** True Postives + True Negatives / Sample Size
- **Recall:** True Positives / (True Positives + False Negatives)
- **AUC**

Given that we are trying to classify fatalities, we believe the cost of misclassifying a non-fatality as a fatality is not as high as classifying a fatality as a non-fatality. Therefore, although accuracy will still be considered to evaluate the model, higher regard will be given to recall.

Furthermore, according to the literature, AUC is an effective way to summarize the overall diagnostic accuracy of the test (Mandrekar, 2010). When interpreting AUC values, 0.5 suggests no better than random chance, 0.7 - 0.80 is considered acceptable, 08-0.9 is considered excellent and more than 0.9 is considered outstanding (Mandrekar, 2010). With respect to the prior, a model would not be accepted until an AUC score of > 0.7 is achieved.

6.1 Initial Results

An initial model was trained on our base level training data set. Forward AIC was used to find the best model fit with the lowest AIC value. The final fit with the lowest AIC can be seen in Appendix D-1.

Initial inspection of the model, the model boasts a 98% accuracy. However, looking closer at the confusion matrix, the model almost exclusively predicts non fatalities (0, 0) and wrongfully predicts 4 fatalities (1, 0) while having 1029 false negatives (0, 1) that were fatalities. Furthermore, having a recall and specificity no much large than 0, the calculated AUC value of 0.5 can be shown by Figure 14, demonstrating the inadequacy of this model.

6.2 What we did to improve

Two primary techniques were trialed to our baseline model to improve its predicting ability as:

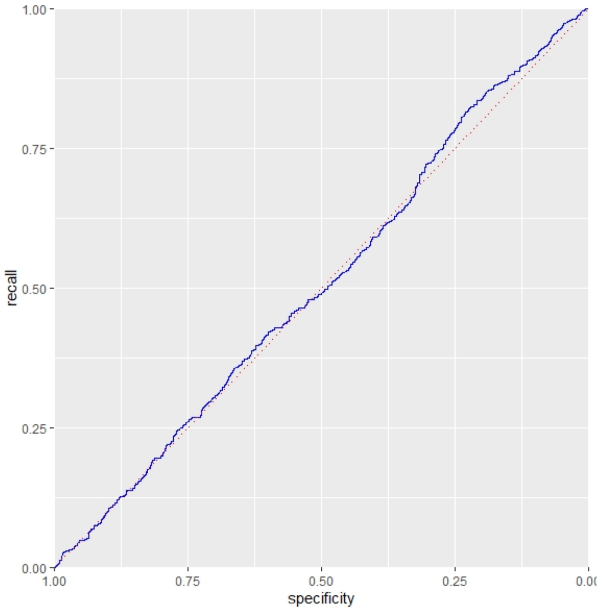

Figure 14: AUC Baseline Model

6.2.1 Balanced Class Training Sets

An imbalance in class in our target variable needed transformations to our training data sets to be performed. The training data is essentially like the notes or textbook a student uses to study for a test. If the student's references do not have information on a particular subject when they come to the exam to find that content, they will perform poorly.

Similarly, if a model has not seen enough instances of a particular class within its training set, it will not know how to identify it. Therefore, random sampling techniques were used to create two new training sets. An under sampled subset where the majority class was reduced and an oversampled subset was created where the minority class was randomly duplicated.

6.2.2 Lift & Reduce Classification Threshold

The default threshold for classing the predicted response as Fatality is >= 0.5. Some arbitrary thresholds were chosen and tested across the three unique training sets. The rationale is that the probability of a fatality occurring may be lower than the default value of 0.5 and therefore the model is misclassifying fatalities as non-fatalities due to predicted probability being too low.

By lowering the threshold, it gives the opportunity to correctly classify those samples that are appearing as false negatives within our confusion matrix. However, this must be done with care to not lower the threshold to the point of which the model overcompensates and classifies true negatives as false positives. Table 5 shows all the variations of model fits that were trialed with the different training sets and threshold lift and reduction.

Model	Accuracy	Precision	Recall	Specificity	AUC	AIC
Base Model	0.98	0	0	0.99	0.5	35130
Under Sample Fatality > 0.50	0.71	0.03	0.72	0.7	0.71	8861
Under Sample Fatality > 0.65	0.8	0.04	0.59	0.8	0.69	8861
Under Sample Fatality > 0.60	0.6	0.03	0.8	0.6	0.7	8861
Under Sample Fatality > 0.45	0.85	0.05	0.45	0.86	0.66	22160
SMOTE > 0.50	0.9	0.05	0.326	0.91	0.62	22160
SMOTE > 0.20	0.8	0.42	0.54	0.81	0.67	22160

Table 6. Model results summary

The final chosen model was "Under Sample Fatality > 0.5". This model was training on the under sampled data set while keeping the classification threshold at 50%. As can be seen in Table 6, this model correctly classified true positives substantially better than the base model while still maintaining to correctly classify true negatives. Furthermore, the recall of 0.72 is greater than any other of the models developed and as stated previously within this report, there is a high value to this metric due to the nature of the classification problem. Finally, the AUC of 0.71 falls within the acceptable range previously identified. For these reasons, this model was accepted as the final model.

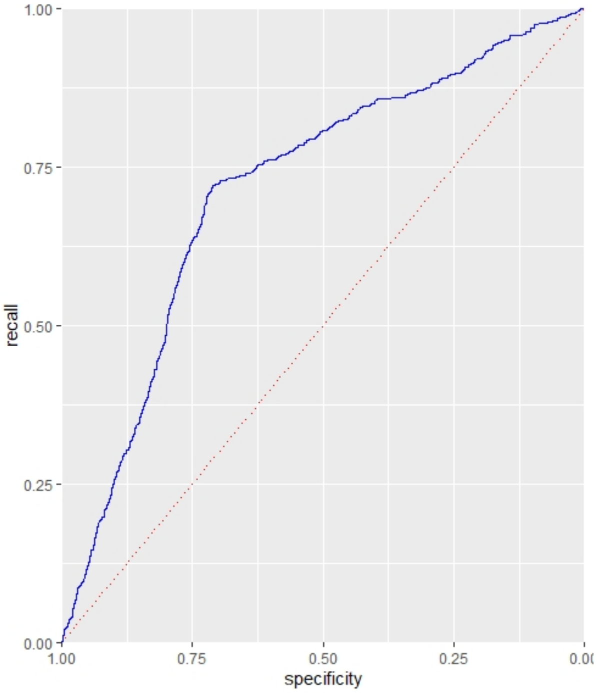

Figure 15: Base model AUC chart

6.3 Evaluation Of Final Model

Once all model variations were tested and reviewed, the best performing model was chosen and then ran on the cross-validation hold-out set (Figure 15). The purpose of the hold out is to test the model on data it has not seen to remove overfitting bias in the training stage (Appendix D-4).

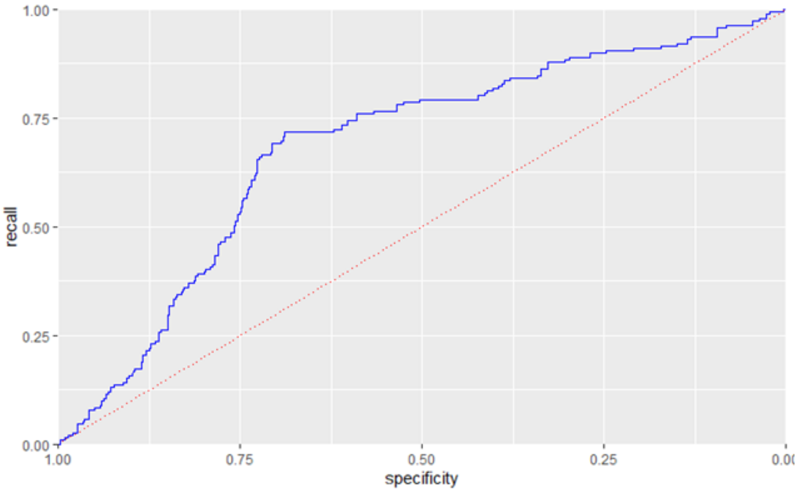

Figure 16: Cross Validation ROC Curve

7 Limitations

7.1 Ethical issues

Lack of data due to ethical issues. This includes personal information like personal attributes of people, such as hobbies, habits and ethnicity. With this data, though sensitive, an effective preventive to target education of certain groups of people for safer driving can be considered. Lack of traffic data

No traffic data of cars traveling that did not get into an accident. The fatalities that we are predicting is if there is an accident.

7.2 Missing data

There were various missing data that we had to use a couple of techniques to replace. Though with the goal to keep the same proportion or mean to reduce the impact on analysis.

7.3 Overall

Limitations of this work were the limited availability of complete data on which to conduct the analysis. So, while the analysis produced non-significance, it is anticipated that as more data becomes available, the models will yield more concrete findings.

Regardless, understanding the relationships among incident causal factors and outcomes may shed light on those causal factors which have the potential to lead to catastrophic events and those which may lead to less severe events.

8 Conclusion

The modeling of road traffic accident data is of high interest for authorities to make better decisions in prioritizing and improving policies and infrastructure as corrective measures. Our study to investigate the fatality causing accidents and their most highly likely combinations of causes will give insights for authorities to make these decisions. To do this we have identified the variables of interest, which are our final features that produces the best model.

This was developed, first, through a process of data collection, cleaning and transformation of data and then several iterations of EDA, developing and evaluation of the model. And finally, to gain valuable insights, understanding the factors that goes in the final model as well as its results is crucial.

To name a few in the order of highest predictability for fatal accidents we found were involved with speed zone, whether accident type struck a pedestrian, struck a fixed object, the gender, the number of people involved and whether atmospheric condition is clear or not.

However, this study is not enough to conclude that these model features are causal to the fatal accident as we have no benchmarking data to compare to. Though, given an accident has occurred these are the prevalent features that are in a fatality. Further research can be looked at these areas to see if there is perhaps a causal relationship.

Regardless, with these insights found, we hope it will help and better equip traffic authorities to make better decisions in prioritizing for corrective measures for road safety.

9 References

1. Australian Automobile Association (2020), Benchmarking the Performance for the National Road Safety Strategy Q4 2019, p. 15.

2. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

3. Hammouri, H. M., Sabo, R. T., Alsaadawi, R., & Kheirallah, K. A. (2020). Handling Skewed Data: A Comparison of Two Popular Methods. Applied Sciences, 10(18), 6247. doi:10.3390/app10186247 https://www.mdpi.com/2076-3417/10/18/6247/htm

4. Murray, C. J., Lopez, A. D., & World Health Organization. (1996). The global burden of disease: a comprehensive assessment of mortality and disability from diseases, injuries, and risk factors in 1990 and projected to 2020: summary. World Health Organization.

5. Nantulya, V. M., & Reich, M. R. (2002). The neglected epidemic: road traffic injuries in developing countries. BMJ (Clinical research ed.), 324(7346), 1139–1141. https://doi.org/10.1136/bmj.324.7346.1139

6. scale function - RDocumentation. (n.d.). RDocumentation. Retrieved October 1, 2021, from https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/scale

7. Transport Accident Comission (2021). Towards Zero 2016-2020 Road Safety Strategy, p.3. Retrieved from http://tac.clients.squiz.net/___data/assets/pdf_file/0010/183556/STU_0206_RS_STRATEGY_2016_web.pdf

8. Watson, W.L. & Ozanne-Smith, J. (1997). The cost of injury to Victoria. Monash University Accident Research Centre, Report No 124.

9. World Health Organization. (n.d.). Road traffic injuries. World Health Organization. Retrieved from https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

10. Yadav, D. (2020). Categorical encoding using Label-Encoding and One-Hot-Encoder. Medium. https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd

11. Christensen, K. (2020). Too much data, too little time - Towards Data Science. Medium. https://towardsdatascience.com/too-much-data-too-little-time-1e7441ecdae1

12. Bhandari, A. (2021). Feature Scaling | Standardization Vs Normalization. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

13. Bursac, Z., Gauss, C. H., Williams, D. K., & Hosmer, D. W. (2008). Purposeful selection of variables in logistic regression. Source code for biology and medicine, 3, 17. https://doi.org/10.1186/1751-0473-3-17

14. Amelia II: A Program for Missing Data. (n.d.). GARY KING. Retrieved from https://gking.harvard.edu/amelia

15. Badr, W. (2019). 6 Different Ways to Compensate for Missing Values In a Dataset (Data Imputation with examples). Medium. Retrieved from https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779

16. Doglun, A., (2020). Module 5: Scan: Outliers. Retrieved from http://rare-phoenix-161610.appspot.com/secured/Module_05.html

17. Maheshwari, Satyam & Jain, R.C. & Jadon, R.s. (2018). An Insight into Rare Class Problem: Analysis and Potential Solutions. Journal of Computer Science. 14. 777-792. 10.3844/jcssp.2018.777.792.

18. Mandrekar, J. N. (2010). Receiver Operating Characteristic Curve in Diagnostic Test Assessment. Journal of Thoracic Oncology, 5(9), 1315–1316. https://doi.org/10.1097/jto.0b013e3181ec173d

19. Agrawal, S. (2021). How to split data into three sets (train, validation, and test) And why? Medium. https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c

20. Saporta, R. (2013, November 28). Understanding `scale` in R. Stack Overflow. https://stackoverflow.com/questions/20256028/understanding-scale-in-r

21. Bill (2014, July 11). What is the reason the log transformation is used with right-skewed distributions? Stack Exchange. https://stats.stackexchange.com/questions/107610/what-is-the-reason-the-log-transformation-is-used-with-right-skewed-distribution

22. Geller, S. (2019, April 5). Normalization vs Standardization — Quantitative analysis. Towards Data Science. https://towardsdatascience.com/normalization-vs-standardization-quantitative-analysis-a91e8a79cebf

23. Lakshmanan S. (2019, May 16). How, When, and Why Should You Normalize / Standardize / Rescale Your Data? Towards AI. https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

24. Vidhya, A. (2020, July 5). Imbalanced Classification Problems in R. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/

25. Pykes, K. (2020, September 10). Oversampling and Undersampling - Towards Data Science. Towards Data Science. https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf

10 Appendix

10.0.1 Appendix A. Tables

10.0.2 Appendix A-1. Accident Data Tables Description

Table	Description
accident	“basic accident details, time, severity, location”
person	“person based details, age, sex etc”
vehicle	“vehicle based data, vehicle type, make etc”
accident_event	“sequence of events e.g. left road, rollover, caught fire”
road_surface_cond	“whether road was wet, dry, icy etc”
atmospheric_cond	“rain, winds etc”
sub_dca	detailed codes describing accident
accident_node	master location table - NB subset of accident table
Node	Lat/Long references

10.0.3 Appendix A-2. Accident Data Attributes Description

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
ACCIDENT_DATE	Text	255	Accident Date	dd/mm/yyyy. (e.g.: 10 July 1995 = 10/07/1995)
ACCIDENT_TIME	Text	255	Accident Time	hh.mm.ss
STAT_DIV_NAME	Text	40	STAT_DIV_NAME is a character field indicating the Metro Melbourne or Country region where the crash occurred.	“Metro, Country.”
ACCIDENT_NO	Text	12	“From November 2005 the accident number field was changed to be 12 character field, starting with T (for example, T20060123456) Where characters 2 to 5 are the year in which accident was registered; Where characters 6 to 12 are a numeric sequencing numbers”	“Example: 12001012345, T20060006259”
ACCIDENTDATE	Date		Date of accident. Australian format DD/MM/YYYY	(e.g.: 10 July 1995 = 10/07/1995)
ACCIDENTTIME	Text	225	“hh.mm.ss. Original date stored in 24 hour format (ie 1pm = 1300 hours) Note the common practice used by the Police, when originally coding up the accident details, of ‘rounding off the time’ to the nearest 5 minutes or even nearest hour. This naturally occurs because in the vast majority of accidents police arrive at the scene well after the accident occurred and so the ‘REAL’ time of the accident is never precisely known.”	Examples of various PC time formats: 24 Hour format 2:35:00 PM = 14:35 or 12 Hour format 2:35:00 PM = 02:35PM 9999 Unknown time midnight = 00:00
ACCIDENT_TYPE	Number		“the type of accident. It is a basic description of what occurred, based on nine categories.”	1 Collision with vehicle 2 Struck pedestrian 3 Struck animal 4 Collision with a fixed object

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
				5 Collision with some other object
				6 Vehicle overturned (no collision)
				7 Fall from or in moving vehicle
				8 No collision and no object struck
				9 Other accident
DAY_OF_WEEK	Number		the day of the week upon which the accident occurred	1 Sunday
				2 Monday
				3 Tuesday
				4 Wednesday
				5 Thursday
				6 Friday
				7 Saturday
DCA_CODE Part 1	Text	3	the Definitions for Classifying Accidents	
LIGHT_CONDITION	Number		the light condition or level of brightness at the time of the accident.	1 Day
				2 Dusk/dawn
				3 Dark street lights on
				4 Dark street lights off
				5 Dark no street lights
				6 Dark street lights unknown
				9 Unknown
NO_PERSONS	Number	4	the number of people involved in the accident	
NO_PERSONS_KILLED	Number	4	Number of people with a given injury level	
NO_PERSONS_INJ_2	Number	4	Number of people with a given injury level	
DCA_CODE Part 2	Text	3		
NO_PERSONS_INJ_3	Number	4	Number of people with a given injury level	
NO_PERSONS_NOT_INJ	Number	4	the number of people that were not injured in the accident	

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
NO_OF_VEHICLES	Number	4	“the number of vehicles involved in the accident. Includes bicycles but not objects, property, toys (skate boards), etc.”	
POLICE_ATTEND	Number		Whether or not the police attended the scene of the accident.	1 Yes 2 No 9 Not known
ROAD_GEOMETRY	Number		The layout of the road where the accident occurred	1 Cross intersection 2 ‘T’ Intersection 3 ‘Y’ Intersection 4 Multiple intersections 5 Not at intersection 6 Dead end 7 Road closure 8 Private property 9 Unknown
SEVERITY	Text		VicRoads estimation of the severity or seriousness of the accident	1 Fatal accident 2 Serious injury accident 3 Other injury accident 4 Non injury accident
DIRECTORY	Text		indicates the name of the street directory used to provide a map reference for the accident.	MEL Melway directory VCD Vic Roads directory
EDITION	Text	70	the edition or version of the street directory used to provide a map reference for the accident	MEL Melway directory VCD Vic Roads directory e.g. ED30

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
PAGE	Text	70	the page number of the street directory used to provide a map reference for the accident	MEL Melway directory VCD Vic Roads directory e.g. 91A
GRID_REFERENCE_X	Text	70	the grid reference in the x direction of the cell in the street directory used to provide a map reference for the accident.	
GRID_REFERENCE_X	Text	70	the grid reference in the y direction of the cell in the street directory used to provide a map reference for the accident.	
SPEED_ZONE	Text	3	the speed zone at the location of the accident. The speed zone is generally assigned to the main vehicle involved.	040 40 km/hr 050 50 km/hr 060 60 km/hr 075 75 km/hr 080 80 km/hr 090 90 km/hr 100 100 km/hr 110 110 km/hr 777 Other speed limit “888 Camping grounds, off road” 999 Not known
NODE_ID	Text	70	The node id of the accident. It starts with 1 and incremented by one when a new accident location is indentified.	e.g. 43078
EVENT_SEQ_NO	Number	4	It starts with 1 and incremented for more than one event in the same accident.	
EVENT_TYPE	Text	1	type of incident event	0 Not applicable 1 Rollover on/off carriageway 2 Fell from vehicle 3 Ran off carriageway

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
				4 Mechanical failure 5 Struck by stone/projectile/load 6 Fell in vehicle 8 Other 9 Not known C Collision
VEHICLE_1_ID	Text	1	first vehicle involved in the event	
VEHICLE_1_COLL_PT	Text	1	collision point on the vehicle.	0 Towed unit 1 Right front corner 2 Right side (forwards) 3 Right side (rearwards) 4 Right rear corner 5 Left front corner 6 Left side (forwards) 7 Left side (rearwards) 8 Left rear corner 9 Not known or Not Applicable F Front N None R Rear S Sidecar T Top/Roof U Undercarriage
VEHICLE_2_ID	Text	1	second vehicle involved in the event.	
VEHICLE_2_COLL_PT	Text	1	collision point on the vehicle.	0 Towed unit 1 Right front corner 2 Right side (forwards) 3 Right side (rearwards) 4 Right rear corner 5 Left front corner 6 Left side (forwards) 7 Left side (rearwards) 8 Left rear corner 9 Not known or Not Applicable F Front N None R Rear S Sidecar T Top/Roof U Undercarriage
PERSON_ID	Text	2	person involved in the specific accident event	

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
OBJECT_TYPE	Text	2	object involved in the specific accident event	1 Pole (telephone/electricity) 2 Tree (shrub/scrub) 3 Fence/Wall (including gates) 17 Traffic island
COMPLEX_INT_NO	Number	4	"the segment is part of a complex intersection. If accident is located in complex intersection, the field has non zero value."	0 Not part of a complex intersection 1-n Valid complex intersection number
ROAD_ROUTE_1	Number	4	This is the primary road/route number for road_name_1.	Group Classifications are: 2000-2999 Freeways or Highways 3000-3999 Forest Rds 4000-4999 Tourist Rds 5000-5999 Main Rds 7000-7999 Ramps (mainly Freeway ramps) 9999 Unclassified Roads e.g. Council / Local roads
ROAD_NAME	Text	45	highest priority road at intersection OR road on which accident took place.	
ROAD_TYPE	Text	15	type of Road_Name	
ROAD_NAME_INT	Text	45	the primary name of the intersecting road	
ROAD_TYPE_INT	Text	15	the type or suffix of the intersecting road	
DISTANCE_LOCATION	Number	4	the distance (in metres) of the accident from the nearest intersecting road (if the crash is a non-intersection or mid-block accident).	Eg: 153
DIRECTION_LOCATION	Text	2	the direction of the accident from the nearest intersecting road (if the crash is a non-intersection or mid-block accident)	N North
				NE North East
				E East
				SE South East

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
				S South SW South West W West NW North West UK Not known
NEAREST_KM_POST	Number	4	the distance (in metres) of the accident to the nearest or closest kilometre post	
NEAREST_KM_POST	Number	4	the distance (in metres) of the accident to the nearest or closest kilometre post	
OFF_ROAD_LOCATION	Text	40	the name of the closest landmark or marker to the accident	
ATMOSPH_COND	Text	1	atmospheric condition	1 Clear 2 Raining 3 Snowing 4 Fog 5 Smoke 6 Dust 7 Strong winds 9 Not known
ATMOSPH_COND_SEQ	Number	4	1 and incremented by 1 if more than one atmospheric condition is entered for the same incident	
LONGITUDE	Double	8	Geographical coordinates	
LATITUDE	Double	8	Geographical coordinates	
NODE_TYPE	Number	1	location type identified by the RCIS spatial system	I Intersection N Non-Intersection O Off Road U Unknown

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
AMG_X	Double	8	“AMG coordinate X value. With the emergence of digital mapping (mid 1980s), the (then) Lands Department of Victoria defined a projection which would allow Victoria to be viewed as a single, continuous map coverage, rather than as multiple zones. This projection, known in VicRoads as Pseudo AMG, is based on AGD 66, but uses a UTM modified to have scale distortion of 1.0 at its centre, a centre based on 145 degrees longitude (Melbourne) and a single zone covering the whole state.”	e.g. 2519154.655
AMG_Y	Double	8	“AMG coordinate Y value. With the emergence of digital mapping (mid 1980s), the (then) Lands Department of Victoria defined a projection which would allow Victoria to be viewed as a single, continuous map coverage, rather than as multiple zones. This projection, known in VicRoads as Pseudo AMG, is based on AGD 66, but uses a UTM modified to have scale distortion of 1.0 at its centre, a centre based on 145 degrees longitude (Melbourne) and a single zone covering the whole state.”	e.g. 2390265.155
LGA_NAME	Text	25	the LGA name	e.g. DANDENONG
SEX	Text	1	the sex or gender of the person	M Male F Female U Not known
AGE	Number	4	how old the person was at the time of the accident. It is calculated by subtracting the person’s birth date from the accident date to give the person’s age in years	

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
INJ_LEVEL	Text	1	the level or degree of injury that the person has experienced as a result of the accident. It is calculated field using inj_police_level and taken_hospital	1 Fatality 2 Serious injury 3 Other injury 4 Not injured
SEATING_POSITION	Text	2	where the person was located on the vehicle	CF Centre-front CR Centre-rear D Driver or rider LF Left-front LR Left-rear NA Not applicable NK Not known OR Other-rear PL Pillion passenger PS Motorcycle sidecar passenger RR Right-rear
HELMET_BELT_WORN	Text	1	whether or not the person was wearing a helmet or seatbelt at the time of the accident	1 Seatbelt worn 2 Seatbelt not worn 3 Child restraint worn 4 Child restraint not worn 5 Seatbelt/restraint not fitted 6 Crash helmet worn 7 Crash helmet not worn 8 Not appropriate 9 Not known

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
ROAD_USER_TYPE	Text	2	the role of the person was at the time of the accident. It is calculated field using person_status and vehicle_type from vehicle table	1 Pedestrian 2 Driver (of V-type 1-9 17 60-63 70-71) 3 Passenger (of V-type 1-9 17 60-63 70-71) 4 Motorcyclist 5 Pillion Passenger 6 Bicyclist (incl. passengers) 7 Other driver (V-type 14-16 99) 8 Other passenger (V-type 14-16 99) 9 Not known
LICENCE_STATE	Text	1	the state of issue of the person's driver license	A Australian Capital Territory B Commonwealth D Northern Territory N New South Wales O Overseas Q Queensland S South Australia T Tasmania V Victoria W Western Australia Z Not known _ Not available (Blank value entered)

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
PEDEST_MOVEMENT	Text	1	“indicates the movement or travel of the person, if classified as a pedestrian”	0 Not applicable 1 Crossing carriageway 2 Working/playing/lying or standing on carriageway 3 Walking on carriageway with traffic 4 Walking on carriageway against traffic 5 Pushing or working on vehicle 6 Walking to/from or boarding tram 7 Walking to/from or boarding other vehicle 8 Not on carriageway (e.g. footpath) 9 Not known
POSTCODE	Number	4	the postcode where the owner of the vehicle resides	
TAKEN_HOSPITAL	Text	1	whether or not the person was taken to hospital	Y Yes N No _ Not Known
EJECTED_CODE	Text	1	whether or not the person was ejected or thrown out of the vehicle	0 Not applicable 1 Total ejected 2 Partially ejected 3 Partial ejection involving extraction

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
SURFACE_COND	Text	1	road surface condition	_ Not known 1 Dry 2 Wet 3 Muddy 4 Snowy 5 Icy 9 Unknown
SURFACE_COND_SEQ	Number	4	starts with 1 and incremented by 1 if more than one road surface condition is entered for the same incident.	
SUB_DCA_CODE	Text	3	SUB_DCA code of the accident. Link to DCA Chart and Sub DCA Codes https://vicroads-public.sharepoint.com/InformationAccess/Shared%20Documents/Road%20Safety/Crash/Accident/DCA_Chart_and_Sub_DCA_Codes.PDF	
SUB_DCA_CODE	Number	4	starts with 1 and incremented by 1 if more than one sub_dca is entered for the same incident Link to DCA Chart and Sub DCA Codes https://vicroads-public.sharepoint.com/InformationAccess/Shared%20Documents/Road%20Safety/Crash/Accident/DCA_Chart_and_Sub_DCA_Codes.PDF	
VEHICLE_YEAR_MANUF	Number	4	indicates the year in which the vehicle was built or manufactured. The data is stored in yyyy format.	
VEHICLE_DCA_CODE	Text	1	“links the vehicle with the movement depicted in the DCA table. For example, if the DCA code for the accident is 111 and the vehicle DCA code is 2, then an inspection of the DCA chart will show that the second vehicle involved in the accident was turning right.”	1 Vehicle 1

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
				2 Vehicle 2 3 Not known which vehicle was number 1 8 Not involved in initial event
INITIAL_DIRECTION	Text	2	“the initial or first direction of travel of the vehicle. For a vehicle that is turning, the initial direction will be different to the final direction. For a non-turning vehicle, the initial direction will be the same as the final direction.”	E East N North NE North east NW North west S South SE South east SW South west W West NK Not known
ROAD_SURFACE_TYPE	Text	1	Prior to 1990 only one road surface was stored. This value is stored with the first vehicle. Road surface for 1990 is available for each vehicle in the collision.	1 Paved 2 Unpaved 3 Gravel 9 Not known
REG_STATE	Text	1	the state which is the vehicle is registered in	A Australian Capital Territory B Commonwealth D Northern Territory N New South Wales O Overseas Q Queensland S South Australia T Tasmania V Victoria W Western Australia Z Not known _ (Blank value entered)/Not available
VEHICLE_BODY_STYLE	Text	6	the body type of the vehicle	

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
VEHICLE_MAKE	Text	6	the vehicle make or manufacturer	
VEHICLE_MODEL	Text	6	the model of the vehicle	E.g. FALCON 0 Unknown 66 Sleeper 75 Tow
VEHICLE_POWER	Number	4	“the power of the vehicle, in CCs or horsepower. For motor cycles, motor scooters and mopeds, the units will be CCs and for all other vehicles the units are rated horsepower.”	0 Unknown 1-1000 Horsepower 1-9999 CCs
VEHICLE_TYPE	Text	2	the type or category of vehicle	
VEHICLE_WEIGHT	Number	4	the weight or mass of the vehicle. The unit of measurement is kilograms.	
CONSTRUCTION_TYPE	Text	1	the construction or formation of the vehicle	A Articulated P Interpretation is not known R Rigid _ (Blank value entered) Unknown
FUEL_TYPE	Text	1	the type of fuel used by the vehicle	D Diesel E Electric G Gas M Multi P Petrol R Rotary Z Unknown
NO_OF_WHEELS	Number	4	the number of wheels that the vehicle has	
NO_OF_CYLINDERS	Number	4	the number of engine cylinders that the vehicle has	
SEATING_CAPACITY	Number	4	the number of seats in the vehicle	
TARE_WEIGHT	Number	4	the tare or unladen weight of the vehicle. The unit of measurement is kilograms	

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
TOTAL_NO_OCCUPANTS	Number	4	indicates the number of occupants or people in the vehicle at the time of the accident	
CARRY_CAPACITY	Number	4	the carry or load capacity of the vehicle. The unit of measurement is kilograms	
CUBIC_CAPACITY	Number	4	indicates the cubic capacity of the engine of the vehicle. The unit of measurement is cubic centimetres	
FINAL_DIRECTION	Text	2	“the final or last direction of travel of the vehicle. For a vehicle that is turning, the initial direction will be different to the final direction. For a non-turning vehicle, the initial direction will be the same as the final direction”	E East
				N North
				NE North east
				NW North west
				S South
				SE South east
				SW South west
				W West
				NK Not known
FINAL_DIRECTION	Text	2	what the driver of the vehicle was attempting to undertake at the time of the accident. This information is meant to obtain via an interview of the vehicle’s driver.	
VEHICLE_MOVEMENT	Text	2	the actual movement of the vehicle prior to the accident.	
TRAILER_TYPE	Text	1	“the type of trailer towed by the vehicle involved in the accident, as reported by the police.”	
VEHICLE_COLOUR_1	Text	3	the primary or main colour of the vehicle.	
VEHICLE_COLOUR_1	Text	3	the secondary colour of the vehicle	

FIELD NAME	TYPE	WIDTH	DEFINITION	DOMAIN
CAUGHT_FIRE	Text	1	whether or not the vehicle caught fire as a result of the accident.	0 Not applicable 1 Yes 2 No 9 Not known
INITIAL_IMPACT	Text	1	the position on the vehicle where the initial impact occurred.	
LAMPS	Text	1	whether the lamps or headlights for the vehicle (under the ambient lighting conditions) were alight (on).	0 Not applicable 1 Yes 2 No 9 Not known
LEVEL_OF_DAMAGE	Text	1	the damage level of the vehicle.	1 Minor 2 Moderate (driveable vehicle) 3 Moderate (unit towed away) 4 Major (unit towed away) 5 Extensive (unrepairable) 6 Nil damage 9 Not known
OWNER_POSTCODE	Number	4	the postcode where the owner of the vehicle resides.	

10.0.4 Appendix A-3. Weather API Data Attributes Description

Domain	Description
Address	"is the address, partial address or latitude,longitude location for which to retrieve weather data. You can also use US ZIP Codes."
Date time	"ISO formatted date, time or datetime value indicating the date and time of the weather data in the locale time zone of the requested location"
Minimum Temperature	minimum temperature at the location.
Maximum Temperature	maximum temperature at the location.
Temperature	temperature at the location
Dew Point	dew point temperature
Relative Humidity	relative humidity in %
Heat Index	" a value between 0 and 10 indicating the level of ultra violet (UV) exposure for that hour or day. 10 represents high level of exposure, and 0 represents no exposure. The UV index is calculated based on amount of short wave solar radiation which in turn is a level the cloudiness, type of cloud, time of day, time of year and location altitude. Daily values represent the maximum value of the hourly values."
Wind Speed	average wind speed over a minute
Wind Gust	instantaneous wind speed at a location – May be empty if it is not significantly higher than the wind speed.
Wind Direction	direction from which the wind is blowing
Wind Chill	
Precipitation	the amount of precipitation that fell or is predicted to fall in the period
Precipitation Cover	the proportion of hours where there was non-zero precipitation
Snow Depth	the depth of snow on the ground
Visibility	distance at which distant objects are visible
Cloud Cover	how much of the sky is covered in cloud ranging from 0-100%
Sea Level Pressure	the sea level atmospheric or barometric pressure in millibars (or hectopascals)
Weather Type	
Latitude	"is the address, partial address or latitude,longitude location for which to retrieve weather data. You can also use US ZIP Codes."
Longitude	"is the address, partial address or latitude,longitude location for which to retrieve weather data. You can also use US ZIP Codes."
Resolved Address	"is the address, partial address or latitude,longitude location for which to retrieve weather data. You can also use US ZIP Codes."
Name	"is the address, partial address or latitude,longitude location for which to retrieve weather data. You can also use US ZIP Codes."

Domain	Description
Info	NA
Conditions	textual representation of the weather conditions.

10.0.5 Appendix A-4. Dealing with missing values techniques

Accident attributes	
X	The ID for the record
ACCIDENT_NO	The accident id that the person was associated with
FATAL_ACCIDENT	Categorical variable on whether the person was involved in an accident with fatalities. This will be our target variable.
ACCIDENTDATE	The date of the accident
ACCIDENTTIME	The time of the accident
DAY_OF_WEEK	The day of week in numerical form
Day_Week_Description	The day of week in categorical form
NO_OF_VEHICLES	The number of vehicles involved in the accident
NO_PERSONS	The number of people involved in the accident
NO_PERSONS_INJ_2	The number of people with an injury level of 2
NO_PERSONS_INJ_3	The number of people with an injury level of 3
NO_PERSONS_KILLED	The number of people died
NO_PERSONS_NOT_INJ	The number of people not injured
Accident_Type_Desc	Description of the accident type
DCA_Description	" ‘Definition for Coding Accidents’ code description. Basically, the category of the accident. "
SEVERITY	The severity of the accident in numerical form
Accident location and environmental attributes	
SPEED_ZONE	The speed zone of where the accident occurred
Road_Geometry_Desc	The road geometry description
ROAD_NAME	Name of the road the accident occurred on
ROAD_TYPE	Type of the road the accident occurred on
ROAD_NAME_INT	Name of the road intersection the accident is closest to
ROAD_TYPE_INT	Type of the road intersection the accident is closest to
LGA_NAME	The local government area name
Road_Surface_Type_Desc	The road surface type description
Surface_Cond_Desc	The road surface condition
LIGHT_CONDITION	The light condition in numerical form
Light_Condition_Desc	The light condition in categorical form
Weather attributes	
Atmosph_Cond_Desc	The atmosphere condition description
Temperature	The average temperature throughout the day

Accident attributes	
DewPoint	The average dewpoint throughout the day
RelativeHumidity	The average relative humidity throughout the day
WindSpeed	The average windspeed throughout the day
Precipitation	The total precipitation throughout the day
Visibility	The average visibility throughout the day. The distance that can seen in daylight.
CloudCover	The average cloud cover throughout the day
WeatherType	The weather types throughout the day

10.0.6 Appendix A-5. Dealing with missing values techniques

Techniques	Definition
Drop / Remove all missing values	The easiest method of dealing with missing values. This generally will exclude all those missing values out of the dataset by removing/deleting column rows.
Imputation Using Mode Values	“Mode/Most frequent is another statistical strategy to impute missing values. It works with categorical features (strings or numerical representations) by replacing missing data with the most frequent values within each column (Badr, 2019).”
Imputation Using (Mean/Median) Values	“A widely common and accepted approach to missing data is to replace the NA with the mean. The nature of this method does not distort or change the distribution of values within each feature. Furthermore, the simplicity to implement this was the driving factor in this method being chosen (Badr, 2019).”
Amelia predictive model (Multiple Imputation)	“Multiple imputation involves imputing values for each missing cell in your data matrix and creating “completed” data sets. Across these completed data sets, the observed values are the same, but the missing values are filled in with a distribution of imputations that reflect the uncertainty about the missing data. (Amelia II: A Program for Missing Data, n.d.)."
K-nearest neighbor	The k nearest neighbors is an algorithm that is used for simple classification. The algorithm uses ‘feature similarity’ to predict the values of any new data points. “This is useful in making predictions about the missing values by finding the k’s closest neighbors to the observation with missing data and then imputing them based on the non-missing values in the neighborhood (Badr, 2019).”
Random Imputation	“On the other hand, replace missing numerical data with mean and replace missing categorical data at existing distribution are the two techniques we chose to deal with missing values in our dataset.”

10.0.7 Appendix A-6. Assign Dummy Variables Techniques

Techniques	Definition		
Label encoding	"A popular encoding technique for categorical information that is used in many applications. In this technique	a unique integer is assigned to each label depending on the alphabetical order of the labels.	
However	depending upon the data values and type of data	label encoding induces a new problem since it uses number sequencing. The problem using the number is that they introduce relation/comparison between them when in reality	there is no relation between these values. "
One Hot Encoding	"This technique can solve this problem by converting each category value into a new column and assigned a 1 or 0 (notation for true/false) value to the column.		
Though this approach eliminates the hierarchy/order issues but does have the downside of adding more columns to the data set. It can cause the number of columns to expand greatly if you have many unique values in a category column (Yadav	2019)."		

Appendix B. Exploratory Data Analysis

```r
library(tidyverse) # data manipulation
library(here) # allocate file
library(dplyr) # data manipulation
library(ggplot2) # data visualization
library(lubridate)
library(patchwork) # merge visual plots to one
library(VIM) # tools for the visualization of missing or imputed values
library(naniar)
library(fastDummies) # automatically create dummy variables columns
library(zoo) # assign mean to NaN values
library(sqldf) # using SQL
library(viridis) # best. color. palette. evar.
library(reshape2)
library(ggrepel)
library(forcats)
library(scales)
library(treemapify) #plot treemap visualization
library(janitor)

#--------------------------------------------------
# Load the dataset
#--------------------------------------------------
df <- read_csv(here('data', 'car_accident.csv'))

## filter necessary columns for analysis
df <- df[-c(1:2,6,8:10,12,18:21,32,34:35,40)]
glimpse(df)
#----------------------------------------------------
# Data Wrangling
#----------------------------------------------------
# get column names
colnms <- colnames(df)

# Create a variable that only contain the columns with missing values
dfNA <- df[ , colSums(is.na(df))!=0]

# Check for NA values
missing_data <- summary(aggr(dfNA,prop=TRUE,combined=TRUE,
                             cex.axis=0.4, sortVars=TRUE))

#------------------------------------------------------------
# Deal with missing data
#------------------------------------------------------------
# categorical variables
df[c(6,17:18,21:22,35:36)] <- df[c(6,17:18,21:22,35:36)]%>%
  replace(is.na(.), "Unknown")

df[c(15,16,19,20)] <- df[c(15,16,19,20)] %>%
  replace(is.na(.), 0)

# numeric variables
```

```
df[c(25,28:34)] <-
  lapply(df[c(25,28:34)], as.numeric) # convert columns to numeric

df[c(25,28:34)] <-
  na.aggregate(df[c(25,28:34)]) # replace NA values with mean

# filter driver seat and injury level
fatality_accidents <- df %>% filter(SEATING_POSITION=="D",
                                    Inj_Level_Desc=="Fatality")

#------------------------------------------------------------
# Data Visualization
#------------------------------------------------------------

#----------------------------------------------------------
# 1. The "When" - Time
#----------------------------------------------------------

####################################################################
## Total Road Fatalities by Year (2006-2020)
####################################################################
# Fatal accident proportion by year
accident_summary_year <- fatality_accidents %>%
  mutate(year = year(ACCIDENTDATE)) %>%
  group_by(year) %>%
  tally()

ggplot(accident_summary_year) +
  aes(x = year, y = n) +
  geom_line(size = 0.5, colour = "#B22222") +
  geom_point(color = "#B22222", size = 2) +
  geom_label(
    aes(label=n),
    nudge_x = 0.5,
    nudge_y = 6,
    check_overlap = TRUE,
    size = 3.5)+
  labs(x = "year",
       y = "total fatalities",
       title = "Total Road Fatalities by Year (2006-2020)") +
  scale_x_continuous(breaks = c(2006, 2008, 2010, 2012,
                                2014, 2016, 2018, 2020)) +
  scale_y_continuous(expand = c(0, 0), limits = c(100, 280),
                     breaks = c(0, 50, 100, 150, 200, 250, 300, 350)) +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))


####################################################################
## Fatal accident proportion by Month (2015-2019)
####################################################################

year_15_20 <- fatality_accidents %>%
```

```r
  filter(ACCIDENTDATE > as.Date("2014-12-31"))

accident_summary_month <-  year_15_20 %>%
  mutate(month = months(as.Date(year_15_20$ACCIDENTDATE))) %>%
  group_by(month) %>%
  tally()

accident_summary_month$month <- ordered( # order month chronically
  accident_summary_month$month, levels=c("January","February","March",
                                         "April","May","June","July",
                                         "August","September","October",
                                         "November","December"))
## bar plot
accident_summary_month %>%
  group_by(month)  %>%
  ggplot(aes(x = month, y = n)) +
  geom_col(fill = "#B22222") +
  ggtitle("Fatalities by Month (2015-2019)") +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 120,140),
                     breaks = c(0,20,40,60,80,100,120,140)) +
  geom_label(aes(x = month, y = n, label = n)) +
  labs(x = "month", y = "fatalities") +
  coord_flip() +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
## Fatalities by Hour and Weekdays (2015-2019)
####################################################################

## add columns for accident hours
fatality_accidents$ACCIDENT_HOUR <- as.character(format(
  strptime(fatality_accidents$ACCIDENTTIME, "%H:%M"), "%H"))

# order Day Of Week chronically
fatality_accidents$Day_Week_Description <- ordered(
  fatality_accidents$Day_Week_Description, levels=c("Monday", "Tuesday", "Wednesday",
                                                    "Thursday", "Friday", "Saturday",
                                                    "Sunday"))
# create new column to specify weekend and weekday
fatality_accidents$ACCIDENTDATE <- as.Date(fatality_accidents$ACCIDENTDATE)
weekdays1 <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')

fatality_accidents$wDay <- factor(
  (weekdays(fatality_accidents$ACCIDENTDATE) %in% weekdays1),
  levels=c(FALSE, TRUE), labels=c('weekend', 'weekday'))

fatal_week_hour <- sqldf(
  "
  SELECT
      Day_Week_Description,
      ACCIDENT_HOUR,
      wDay,
      COUNT(*) as total
      FROM fatality_accidents
    GROUP BY
      Day_Week_Description,
      ACCIDENT_HOUR,
      wDay"
)

#heatmap
ggplot(fatal_week_hour) +
  aes(
    x = ACCIDENT_HOUR,
    y = Day_Week_Description,
    fill = `total`
  ) +
  geom_tile(size = 1.2) +
  scale_fill_distiller(palette = "Reds", direction = 1) +
  labs(
    x = "hours",
    y = "weekdays",
    title = "Weekday vs. Hourly Road Fatalities (2006-2020)",
    fill = "fatalities"
  ) +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
## Fatalities by Weekend Vs. Weekdays Distribution (2006-2020)
####################################################################
#barplot
ggplot(fatal_week_hour) +
  aes(x = ACCIDENT_HOUR, y = total, fill = wDay) +
  geom_boxplot(shape = "circle") +
  scale_fill_brewer(palette = "Reds", direction = -1) +
  labs(
    x = "hour",
    y = "fatalities",
    title = "Fatalities by Time (2006-2020)"
  ) +
  ggthemes::theme_tufte() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 15L,
                              hjust = 0.5),
    plot.subtitle = element_text(size = 13L,
                                 hjust = 0.5)
  ) +
  facet_wrap(vars(wDay))


####################################################################
## Fatalities by Day and Night (General) (2006-2020)
####################################################################
```

```r
morning_hour <- c('00','01','02','03','04','05',
                  '06','07','08','09','10','11','12')
afternoon_hour <- c('13','14','15','16','17')
evening_hour <- c('18','19','20')
night_hour <- c('21','22','23')

# create new column specify Day vs Night time
fatality_accidents$day_night<-
  ifelse(fatality_accidents$ACCIDENT_HOUR %in% morning_hour, "Morning",
         ifelse(fatality_accidents$ACCIDENT_HOUR %in% afternoon_hour, "Afternoon",
                ifelse(fatality_accidents$ACCIDENT_HOUR %in% evening_hour, "Evening",
                       ifelse(fatality_accidents$ACCIDENT_HOUR %in% night_hour, "Night",NA))))


#create new variable
day_night <- sqldf(
  "
  SELECT
      day_night,
      COUNT(*) as value
      FROM fatality_accidents
    GROUP BY day_night
      "
)

# calculate percentage
day_night %>%
  arrange(desc(value)) %>%
  mutate(prop = percent(value / sum(value))) -> day_night

# pie chart
ggplot(day_night, aes(x = "", y = value, fill = fct_inorder(day_night))) +
  geom_bar(stat = "identity", width = 1) +
  geom_col(color = "black", width = 1) +
  coord_polar("y", start = 0) +
  geom_label_repel(aes(label = prop), size=5,
                   show.legend = F, nudge_x =1, nudge_y = 1) +
  labs(
    title = "Fatalities by Daytime (2006-2020)"
  ) +
  scale_fill_brewer(palette = "Reds") +
  theme_classic() +
  guides(fill = guide_legend(title = "Daytime")) +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
## Fatalities by Day and Night (Weekend Vs. Weekdays) (2006-2020)
####################################################################
#create new variable
day_night_wDay <- sqldf(
  "
  SELECT
```

```
      day_night,
      SUM(CASE WHEN wDay = 'weekend' THEN 1 ELSE 0 END) AS weekend,
      SUM(CASE WHEN wDay = 'weekday' THEN 1 ELSE 0 END) AS weekday
      FROM fatality_accidents
    GROUP BY day_night
      "
)

# Transform the data into the long format
day_night_wDay <- melt(day_night_wDay)

# double pie charts
ggplot(day_night_wDay, aes(x = "", y = value, fill = day_night)) +
  geom_bar(stat = "identity", width = 1, position = position_fill()) +
  coord_polar(theta = "y") +
  facet_wrap( ~ variable) +
  scale_fill_brewer(palette = "Reds") +
  theme_classic() +
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        plot.title = element_text(hjust = 0.5, color = "#666666"))

#----------------------------------------------------------
# 1. The "Where" - Location
#----------------------------------------------------------

####################################################################
## Top 10 LGA with highest road fatalities (2006-2020)
####################################################################

fatality_accidents %>%
  group_by(LGA_NAME) %>%
  dplyr::summarise(Total = n()) %>%
  top_n(10, Total)  %>%
  ggplot(aes(area = Total, fill = Total, label = LGA_NAME)) +
  geom_treemap() +
  labs(
    title = "Top 10 LGA with highest road fatalities (2006-2020)"
  ) +
  geom_treemap_text(fontface = "italic", colour = "white", place = "topleft",
                    reflow = T,grow = TRUE) +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
## Fatalities by Road Geometry (2006-2020)
####################################################################
fatal_geo <- sqldf(
  "
  SELECT
      Road_Geometry_Desc,
      COUNT(*) AS value
      FROM fatality_accidents
      where Road_Geometry_Desc != 'Unknown'
    GROUP BY
      Road_Geometry_Desc
    ORDER BY COUNT(*) DESC
      "
)

# bar plot
ggplot(fatal_geo) +
  aes(x = Road_Geometry_Desc, y = value) +
  geom_bar(stat='identity', fill = "#FF8C00") +
  geom_label(
    aes(x = Road_Geometry_Desc, y = value, label=value))+
  labs(
    x = "road geometry",
    y = "fatalities",
    title = "Fatalities by Road Geometry (2006-2020)"
  ) +
  coord_flip() +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))


#------------------------------------------------------------
# 1. The "Why" - Other factors
#------------------------------------------------------------
######################################################################
## Fatalities by Speed (2006-2020)
######################################################################
#create new variable
fatal_speed <- sqldf(
  "
  SELECT
      SPEED_ZONE,
      Age_Group,
      COUNT(*) AS value
      FROM fatality_accidents
      WHERE SPEED_ZONE NOT IN ('030','075','888','999')
    GROUP BY
      SPEED_ZONE,
      Age_Group
      "
)

# bar plot
ggplot(fatal_speed) +
  aes(x = SPEED_ZONE, y = value) +
  geom_bar(stat='identity',fill="#B22222") +
  labs(
    x = "speed",
    y = "fatalities",
    title = "Fatalities by Speed (2006-2020)"
  ) +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
# Pie chart Road User Type
####################################################################
#create new variable
arranged <- sqldf(
  "
  SELECT
      Road_User_Type_Desc,
      COUNT(*) as value
    FROM fatality_accidents
    WHERE Road_User_Type_Desc != 'Unknown'
    GROUP BY Road_User_Type_Desc
      "
)

# calculate percentage
arranged %>%
  arrange(desc(value)) %>%
  mutate(prop = percent(value / sum(value))) -> arranged

# pie chart
ggplot(arranged, aes(x = "", y = value,
                     fill = fct_inorder(Road_User_Type_Desc))) +
  geom_bar(stat = "identity", width = 1) +
  geom_col(color = "black", width = 1) +
  coord_polar("y", start = 0) +
  geom_label_repel(aes(label = prop),
                   size=5,
                   show.legend = F,
                   nudge_x =1,
                   nudge_y = 1) +
  labs(
    title = "Fatalities by Road Users (2006-2020)"
  ) +
  scale_fill_brewer(palette = "Reds") +
  theme_classic() +
  guides(fill = guide_legend(title = "Road Users")) +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
# Pie chart Accident Type
####################################################################
#create new variable
accident_type <- sqldf(
  "
  SELECT
      Accident_Type_Desc,
      COUNT(*) as value
    FROM fatality_accidents
    WHERE Accident_Type_Desc != 'Unknown'
    GROUP BY Accident_Type_Desc
      "
)

# calculate percentage
accident_type %>%
  arrange(desc(value)) %>%
  mutate(prop = percent(value / sum(value))) -> accident_type

# pie chart
ggplot(accident_type, aes(x = "", y = value,
                          fill = fct_inorder(Accident_Type_Desc))) +
  geom_bar(stat = "identity", width = 1) +
  geom_col(color = "black", width = 1) +
  coord_polar("y", start = 0) +
  labs(
    title = "Fatalities by Accident Types (2006-2020)"
  ) +
  scale_fill_brewer(palette = "Reds") +
  theme_classic() +
  guides(fill = guide_legend(title = "Accident Types")) +
  ggthemes::theme_tufte() +
  theme(plot.title = element_text(size = 15L, hjust = 0.5))

####################################################################
# Number of Fatal accidents by Road Surface Condition
####################################################################
car_accidents <- read_csv(here("data/car_accident.csv"))
car_accidents <- clean_names(car_accidents)

df2 <- car_accidents %>%
  filter(seating_position == 'D') %>%
  group_by(surface_cond_desc) %>%
  summarise(total_fatalities = n_distinct(accident_no),
            number_of_sameples = n(),
            fatalities_ratio = total_fatalities / number_of_sameples)

df2 %>%
  filter(surface_cond_desc != 'Unknown') %>%
  ggplot(aes(x = surface_cond_desc, y = fatalities_ratio)) +
  geom_bar(stat = 'identity', fill = "#B22222") +
  xlab("Surface Condition Description") +
  ylab("Total Accidents with Fatalities / Total Accidents") +
  labs(title = "Number of Fatal Accidents by Road Surface Conditions") +
  ggthemes::theme_tufte() +
  theme(
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5)
  )

####################################################################
```

```
# Number of Fatal accidents by Weather Conditions
####################################################################
df3 <- car_accidents %>%
  filter(seating_position == 'D') %>%
  group_by(conditions) %>%
  summarise(total_fatalities = n_distinct(accident_no),
            number_of_sameples = n(),
            fatalities_ratio = total_fatalities / number_of_sameples)


df3 %>%
  filter(conditions != 'Unknown') %>%
  ggplot(aes(x = conditions, y = fatalities_ratio)) +
  geom_bar(stat = 'identity', fill = "#B22222") +
  xlab("Weather Condition Description") +
  ylab("Total Accidents with Fatalities / Total Accidents") +
  labs(title = "Number of Fatal Accidents by Weather Conditions") +
  coord_flip() +
  ggthemes::theme_tufte() +
  theme(
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5)
  )

####################################################################
# Number of Fatal accidents by Light Conditions
####################################################################
df4 <- car_accidents %>%
  filter(seating_position == 'D') %>%
  group_by(light_condition_desc) %>%
  summarise(total_fatalities = n_distinct(accident_no),
            number_of_sameples = n(),
            fatalities_ratio = total_fatalities / number_of_sameples)


df4 %>%
  filter(light_condition_desc != 'Unknown') %>%
  ggplot(aes(x = light_condition_desc, y = fatalities_ratio)) +
  geom_bar(stat = 'identity', fill = "#B22222") +
  xlab("Light Conditions Description") +
  ylab("Total Accidents with Fatalities / Total Accidents") +
  labs(title = "Number of Fatal Accidents by Light Conditions") +
  coord_flip() +
  ggthemes::theme_tufte() +
  theme(
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5)
  )
```

Appendix C. Data Pre-processing

```r
library(tidyverse)
library(here)
library(caret)
library(dplyr)
library(mltools)
library(VIM)
library(summarytools)
library(moments)
library(outliers)
library(DataExplorer)

df <- read_csv(here('data', 'car_accident.csv'))

################################################################################
# Filter and mutate data for basic analysis ----
################################################################################
# Convert fatality to numerical
df <- df %>%
  mutate(FATAL_ACCIDENT =  case_when(
    FATAL_ACCIDENT == 'Y' ~ 1,
    FATAL_ACCIDENT == 'N' ~ 0)
  )

# convert speed zone to numeric
df <- transform(df, SPEED_ZONE = as.numeric(SPEED_ZONE))
# Filter df - filter to driver based data and remove outliers
# drop values where speed zone has incorrect data e.g. > 200km/hr
df <- df[df$SPEED_ZONE < 200, ]
df <- df[df$NO_OF_CYLINDERS < 25, ]

# Convert no_of_cylinders to factor
df$NO_OF_CYLINDERS <- as.factor(df$NO_OF_CYLINDERS)

target <- c('Drivers', 'Motorcyclists')
df <- df %>%
  filter(Road_User_Type_Desc %in% target)

dfSummary(df)


################################################################################
# Define numerical and categorical columns ----
################################################################################

numerical_cols <- c('NO_OF_VEHICLES', 'NO_PERSONS','SPEED_ZONE',
                    'VEHICLE_YEAR_MANUF', 'TOTAL_NO_OCCUPANTS',
                    'LIGHT_CONDITION', 'CloudCover', 'WindSpeed',
                    'Temperature', 'DewPoint', 'RelativeHumidity',
                    'Precipitation')
```

```
cat_cols <- c('SEX', 'SEATING_POSITION', 'Accident_Type_Desc',
              'Road_Surface_Type_Desc', 'Surface_Cond_Desc',
              'Atmosph_Cond_Desc', 'Light_Condition_Desc','Conditions',
              'Age_Group', 'Day_Week_Description', 'NO_OF_CYLINDERS')

indexing_cols <- c('ACCIDENT_NO', 'FATAL_ACCIDENT', 'ACCIDENTDATE',
                   'ACCIDENTTIME' , 'Road_User_Type_Desc')

log_cols <- c('Precipitation', 'CloudCover', 'RelativeHumidity', 'NO_PERSONS',
              'TOTAL_NO_OCCUPANTS', 'NO_OF_VEHICLES', 'SPEED_ZONE',
              'LIGHT_CONDITION')

all_features_cols <- c(cat_cols)
```

Appendix C-1. Random Imputation - Handle Missing Values Technique

```
################################################################################
# Handle missing values-----
################################################################################
################################################################################
# Replace missing values at same frequency they appear in column
################################################################################
# which(myV>7)[1]
idx_greater_than <- function(value, list){
#Find the first index of vector 'list' that has a corresponding value greater
#than 'value'

  for(i in 1:length(list)){
    if(list[i] > value){
      return(i)
    }
  }
}

# MAIN LOGIC STARTS HERE ----
replace_nan_df <- df[all_features_cols]

for(name in names(replace_nan_df)){
  column_vector <- pull(replace_nan_df, name)

  # Get index of nans
  nan_idxs <- which(is.na(column_vector))

  # If no nans, don't worry
  if(length(nan_idxs)==0){
    next
  }

  srs_notnull <- column_vector[!is.na(column_vector)]

  # Get unique labels and counts for the non-nan features
  unique_frequency_df <- as.data.frame(table(srs_notnull))
  labels <- as.character(unique_frequency_df$srs_notnull)
  counts <- unique_frequency_df$Freq
  cum_counts <- cumsum(counts)

  # Generate random numbers of size len(nan_idxs)
  set.seed(1)
  rand_vals <- floor(runif(length(nan_idxs), min=0, max=length(srs_notnull)))

  new_vals <- c()
  for(x in rand_vals){
    #Find out the largest number in cum_counts that each rand_val is less than
    larger_value_index <- idx_greater_than(x, cum_counts)
    # Get values corresponding to above index
    new_vals <- append(new_vals, labels[larger_value_index])
  }
  # Update the df with the new vals
  df[nan_idxs, name] = new_vals
}

sum(is.na(df))
```

Appendix C-2. Log Transformation

```
################################################################################
# NEED TO PERFORM LOG FUNCTION ON SKEWED NUMERICAL DATA HERE
################################################################################
log_df <- df[log_cols]
glimpse(df)
for(name in names(log_df)){
  column_vector <- pull(log_df, name)
  column_vector <- as.numeric(column_vector)
  skew <- skewness(column_vector)
  if (skew < - 0.5){
    constant <- max(column_vector) + 1
    new_vals <- constant - column_vector
    new_vals <- log(new_vals)
    df[name] = as.numeric(new_vals)
    hist(new_vals)
  } else if (skew > 0.5){
    constant <- 1
    new_vals <- log((column_vector + constant))
    new_vals <- as.numeric(new_vals)
    df[name] = new_vals
  }
}
```

Appendix C-3. Transform Categorical Data Based on EDA

```
################################################################################
# Transform Categorical Data Based on EDA ----
```

```
################################################################################
#Use dummyVars function to create binary variables.
category_df <- df[cat_cols]
variables <- dummyVars("~.", data = category_df, sep = "_")


category_df <- data.frame(predict(variables, newdata = category_df))

category_df <- category_df %>%
     mutate_if(is.double, as.factor)

base_df <- df[indexing_cols]
numerical_df <- df[numerical_cols]
```

Appendix C-4. Outlier

```
################################################################################
# Outlier
################################################################################
# calculate z-score
mean(df$AGE)
#calculate z score
z.scores <- df$AGE %>% na.omit %>% scores(type = "z")
z.scores %>% summary()

# Finds the total number of outliers according to the z-score
length (which( abs(z.scores) >3 ))
```

Appendix C-5. Outlier

```
################################################################################
# Outlier
################################################################################
# calculate z-score
mean(df$AGE)
#calculate z score
z.scores <- df$AGE %>% na.omit %>% scores(type = "z")
z.scores %>% summary()

# Finds the total number of outliers according to the z-score
length (which( abs(z.scores) >3 ))
```

Appendix C-6. Standardization

```
################################################################################
# Standardize/ scale numeric values
################################################################################
```

```r
final_df <- cbind(base_df, category_df)
final_df <- cbind(final_df, numerical_df)
```

Appendix C-7. Save Elements For Balancing Data

```r
################################################################################
# Create features to keep in df based on EDA and domain knowledge
################################################################################
final_cols <- c("ACCIDENT_NO","FATAL_ACCIDENT",
                "ACCIDENTDATE","ACCIDENTTIME",
                "Road_User_Type_Desc","SEXF",
                "SEXM",
                "Accident_Type_DescCollision.with.a.fixed.object",
                "Accident_Type_DescStruck.animal",
                "Accident_Type_DescStruck.Pedestrian",
                "Accident_Type_DescVehicle.overturned..no.collision.",
                "Road_Surface_Type_DescUnpaved",
                "Surface_Cond_DescDry",
                "Surface_Cond_DescIcy",
                "Surface_Cond_DescMuddy",
                "Surface_Cond_DescSnowy",
                "Surface_Cond_DescWet",
                "Atmosph_Cond_DescClear",
                "Atmosph_Cond_DescFog",
                "Atmosph_Cond_DescRaining",
                "Atmosph_Cond_DescSmoke",
                "Atmosph_Cond_DescStrong.winds",
                "Light_Condition_DescDark.No.street.lights",
                "Light_Condition_DescDark.Street.lights.off",
                "ConditionsOvercast",
                "ConditionsRain",
                "ConditionsRain..Overcast",
                "Age_Group16.17",
                "Age_Group17.21",
                "Age_Group70.",
                "Day_Week_DescriptionSaturday",
                "Day_Week_DescriptionSunday",
                "NO_OF_CYLINDERS_4",
                "NO_OF_CYLINDERS_6",
                "NO_OF_CYLINDERS_8",
                "NO_OF_CYLINDERS_12",
                "NO_OF_VEHICLES",
                "NO_PERSONS",
                "SPEED_ZONE",
                "VEHICLE_YEAR_MANUF",
                "TOTAL_NO_OCCUPANTS",
                "LIGHT_CONDITION",
                "CloudCover",
                "WindSpeed",
                "Temperature",
                "DewPoint",
                "RelativeHumidity",
                "Precipitation")


final_df <- final_df[final_cols]
dfSummary(final_df)
write_csv(final_df, here("data", "Car_Accident_Data_No_Na.csv"))
```

Appendix C-8. Balancing Data

```
library(tidyverse)
library(here)
library(caret)
library(dplyr)
library(mltools)
library(VIM)
library(summarytools)
library(DMwR)

################################################################################
# Read in data
df <- read.csv(here("data", "Car_Accident_Data_No_Na.csv"))

################################################################################
# Create train, test and cross validation df function
################################################################################
create_train_test_cross <- function(df){
  train_df <- df[df$ACCIDENTDATE <  "2017-01-01",]
  test_df <- df[(df$ACCIDENTDATE > "2017-01-01" & df$ACCIDENTDATE < "2020-01-01"),]
  cross_valid_df <- df[(df$ACCIDENTDATE > "2020-01-01" ),]
  return (list(train_df, test_df, cross_valid_df))
}
################################################################################
################################################################################
# Create different balanced data sets
################################################################################
################################################################################
# Create train and test for no transformation techniques
################################################################################
no_change_list <- create_train_test_cross(df)
no_change_train <- no_change_list[[1]]
no_change_test <- no_change_list[[2]]
no_change_cross <- no_change_list[[3]]

drops <- c("Road_User_Type_Desc","ACCIDENTTIME", "ACCIDENT_NO", "ACCIDENTDATE")
no_change_train <- no_change_train[ , !(names(no_change_train) %in% drops)]
no_change_test <- no_change_test[ , !(names(no_change_test) %in% drops)]
no_change_cross <- no_change_cross[ , !(names(no_change_cross) %in% drops)]

write_csv(no_change_train, here("data", "no_change_train.csv"))
write_csv(no_change_test, here("data", "test_set_generic.csv"))
write_csv(no_change_cross, here("data", "cross_set_generic.csv"))
```

Appendix D. Logistic Regression Model and Evaluation

```
library(MASS)
library(tidyverse)
library(caret)
library(here)
library(pROC)
```

```
knitr::opts_chunk$set(warning = TRUE, message = TRUE)
no_change_train <-read_csv(here("data", "no_change_train.csv"))
test_generic <- read_csv(here("data", "test_set_generic.csv"))
smote_train <- read_csv(here("data", "smote_train.csv"))
under_sample_train <- read_csv(here("data", "under_sample_train.csv"))
cross_validation <- read_csv(here("data", "no_change_cross.csv"))
```

Appendix D-1. Baseline Model

```
base_final_model <- glm(formula = FATAL_ACCIDENT ~ SPEED_ZONE +
                    Accident_Type_DescStruck.Pedestrian +
                    Accident_Type_DescCollision.with.a.fixed.object + SEXM +
                    NO_PERSONS + Light_Condition_DescDark.No.street.lights +
                    Atmosph_Cond_DescClear + Age_Group70. + CloudCover +
                    NO_OF_CYLINDERS_4 +  LIGHT_CONDITION +
                    Accident_Type_DescStruck.animal + ConditionsRain..Overcast +
                    NO_OF_CYLINDERS_12 + Surface_Cond_DescIcy + SEXF +
                    Age_Group17.21 + Atmosph_Cond_DescRaining +
                    Surface_Cond_DescDry + Surface_Cond_DescWet +
                    VEHICLE_YEAR_MANUF +
                    Accident_Type_DescVehicle.overturned..no.collision. +
                    ConditionsRain + Atmosph_Cond_DescFog +
                    Atmosph_Cond_DescStrong.winds + NO_OF_VEHICLES +
                    NO_OF_CYLINDERS_6 + Surface_Cond_DescSnowy +
                    NO_OF_CYLINDERS_8 + Age_Group16.17,
                    family = binomial, data = no_change_train)
```

Appendix D-2. Testing Model

```
model_test <- test_generic[,-1]

pred <- predict(base_final_model, model_test, type = "response")
pred <- as.data.frame(pred)
lift_threshold <- 0.5
pred <- mutate(pred, pred = ifelse(pred >= lift_threshold, 1,
                                   ifelse(pred < lift_threshold, 0, NA)))

pred_y <- as.numeric(pred > 0)
true_y <- as.numeric(test_generic$FATAL_ACCIDENT)
pred_y_factor <- as.factor(pred_y)
```

```
true_y_factor <- as.factor(true_y)
confusion_matrix_1 <- confusionMatrix(pred_y_factor, true_y_factor, positive = "1")
```

Appendix D-3. Confusion Matrix

```
print(confusion_matrix_1)
print(" ")
```

```
print(confusion_matrix_1$byClass)
```

```
#Create ROC curve
idx <- order(-pred)
recall <- cumsum(true_y[idx] == 1) / sum(true_y == 1)
specificity <- (sum(true_y == 0) - cumsum(true_y[idx] == 0)) / sum(true_y == 0)
roc_df <- data.frame(recall = recall, specificity = specificity)
roc <- ggplot(roc_df, aes(x=specificity, y=recall)) +
  geom_line(color='blue') +
  scale_x_reverse(expand=c(0, 0)) +
  scale_y_continuous(expand=c(0, 0)) +
  geom_line(data=data.frame(x=(0:100) / 100), aes(x=x, y=1-x),
            linetype='dotted', color='red')

print(roc)

auc <- sum(roc_df$recall[-1] * diff(1 - roc_df$specificity))
print(paste0("AUC: ", auc))
```

Appendix D-4. Under Sample Final Model

```
under_sample_final_model <- glm(formula = FATAL_ACCIDENT ~ SPEED_ZONE +
                                Accident_Type_DescStruck.Pedestrian +
                                Accident_Type_DescCollision.with.a.fixed.object +
                                SEXM +  NO_PERSONS + Atmosph_Cond_DescClear +
                                LIGHT_CONDITION + Age_Group70. + CloudCover +
                                Atmosph_Cond_DescRaining +
                                Atmosph_Cond_DescStrong.winds +
                                Atmosph_Cond_DescFog + NO_OF_CYLINDERS_4 +
                                Surface_Cond_DescDry + Surface_Cond_DescWet +
                                Accident_Type_DescStruck.animal +
                                VEHICLE_YEAR_MANUF + Age_Group17.21 +
                                ConditionsRain..Overcast +
                                Surface_Cond_DescSnowy + NO_OF_CYLINDERS_6 +
                                Road_Surface_Type_DescUnpaved +
                                Light_Condition_DescDark.No.street.lights,
                                family = binomial,
                                data = under_sample_train)
```

Appendix D-5. ROC Curves

```
model_test <- test_generic[,-1]
pred <- predict(under_sample_final_model, model_test, type = "response")
pred <- as.data.frame(pred)
pred <- mutate(pred, pred = ifelse(pred >= lift_threshold, 1,
                                   ifelse(pred < lift_threshold, 0, NA)))

print("Compiling confusion matrix")
pred_y <- as.numeric(pred > 0)
true_y <- as.numeric(test_generic$FATAL_ACCIDENT)
pred_y_factor <- as.factor(pred_y)
true_y_factor <- as.factor(true_y)
confusion_matrix_2 <- confusionMatrix(pred_y_factor, true_y_factor, positive = "1")
print(confusion_matrix_2)
print(confusion_matrix_2$byClass)

#Create ROC curve
idx <- order(-pred)
recall <- cumsum(true_y[idx] == 1) / sum(true_y == 1)
specificity <- (sum(true_y == 0) - cumsum(true_y[idx] == 0)) / sum(true_y == 0)
roc_df <- data.frame(recall = recall, specificity = specificity)
roc <- ggplot(roc_df, aes(x=specificity, y=recall)) +
  geom_line(color='blue') +
  scale_x_reverse(expand=c(0, 0)) +
  scale_y_continuous(expand=c(0, 0)) +
  geom_line(data=data.frame(x=(0:100) / 100), aes(x=x, y=1-x),
            linetype='dotted', color='red')

print(roc)


auc <- sum(roc_df$recall[-1] * diff(1 - roc_df$specificity))
print(paste0("AUC: ", auc))


print(under_sample_final_model)
```

```
model_test <- cross_validation[,-1]
pred <- predict(under_sample_final_model, model_test, type = "response")
pred <- as.data.frame(pred)
pred <- mutate(pred, pred = ifelse(pred >= lift_threshold, 1,
                                   ifelse(pred < lift_threshold, 0, NA)))

print("Compiling confusion matrix")
pred_y <- as.numeric(pred > 0)
true_y <- as.numeric(cross_validation$FATAL_ACCIDENT)
pred_y_factor <- as.factor(pred_y)
true_y_factor <- as.factor(true_y)
confusion_matrix_3 <- confusionMatrix(pred_y_factor, true_y_factor, positive = "1")
print(confusion_matrix_3)
print(confusion_matrix_3$byClass)
```

```r
#Create ROC curve
idx <- order(-pred)
recall <- cumsum(true_y[idx] == 1) / sum(true_y == 1)
specificity <- (sum(true_y == 0) - cumsum(true_y[idx] == 0)) / sum(true_y == 0)
roc_df <- data.frame(recall = recall, specificity = specificity)
roc <- ggplot(roc_df, aes(x=specificity, y=recall)) +
  geom_line(color='blue') +
  scale_x_reverse(expand=c(0, 0)) +
  scale_y_continuous(expand=c(0, 0)) +
  geom_line(data=data.frame(x=(0:100) / 100), aes(x=x, y=1-x),
            linetype='dotted', color='red')

print(roc)


auc <- sum(roc_df$recall[-1] * diff(1 - roc_df$specificity))
print(paste0("AUC: ", auc))


print(under_sample_final_model)
```