UK road accident analysis and prediction for accident severity and casualties

aditya chandra

MSc in Data Analytics

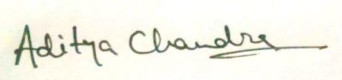
Athlone Institute of Technology



##### Declaration

I hereby certify that the material, which is submitted in this thesis towards the award of MSc. in Data Analytics, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above-named award.

Future students may use the material contained in this thesis provided that the source is acknowledged in full.



Signed\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_18/08/2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_

##### Abstract

Every year thousands of people in the United Kingdom lose their lives due to accidents. According to the 2018 annual report on the reported road casualties in Great Britain issued by the Department for Transport, a total of 1,784 road deaths were reported in 2018 which was similar to the observed level of road deaths since 2012. Therefore, it is essential to analyse the data related to the accidents to understand the relationship between various conditions on the road and the severity of an accident. By recognizing those factors that have the most impact on accidents, people could be more cautious on road and avoid accidents in future. The exploratory data analysis and descriptive analysis of the historical data is aimed towards the identification of these factors and conditions to spread awareness among the commuters.

Through the prediction for the severity of an accident in future, the governing bodies related to the healthcare could provide appropriate service and take necessary actions to reduce the chances of death in road accidents. Moreover, if additional information such as the number of casualties, their gender and average age could be predicted in advance, early and proper medical care could be administered to them in time which could save their lives. The prediction for number of vehicles involved in an accident in future could help the authorities in being prepared for the removal of damaged vehicles from the roads to avoid disruption in the flow of traffic. These analyses could be useful to an array of experts, including [forensic scientists](https://en.wikipedia.org/wiki/Forensic_scientists), [forensic engineers](https://en.wikipedia.org/wiki/Forensic_engineering#Forensic_engineering) or [health and safety](https://en.wikipedia.org/wiki/Health_and_safety) advisers. This research is aimed to analyse Great Britain’s road safety data from 2015 to 2018 issued by the department for transport of the UK to identify the conditions present during most severe accidents and accidents in general with the help of data visualization techniques. The project aims to predict the severity of future accidents, the number of casualties and vehicles related to future accidents, the majority gender of casualties for future accidents and the average age of casualties for future accidents based on the information available from the historical data.

##### Acknowledgements

I would like to thank my project supervisor Mr Barry O'Loughlin, Business Faculty, Athlone Institute of Technology, for his relentless support and guidance. His constant encouragement was invaluable to the completion of this research project.

I would also like to thank Ms Emer Connolly from the Academic Writing Centre at Athlone Institute of Technology for her support and suggestions with the writing section. Her flexibility with the timings ensured that I received all the necessary feedback on time which proved extremely helpful in the writing of this thesis.

Lastly, I would like to extend my gratitude to my family for their continuous encouragement and emotional support without which this accomplishment would not have been possible.

Table of Contents

[Chapter 1 - Introduction 1](#_Toc48567592)

[1.1. Introduction 1](#_Toc48567593)

[1.2. Background 3](#_Toc48567594)

[1.3. Research Aim and Objectives 4](#_Toc48567595)

[Chapter 2 – Literature Review 5](#_Toc48567596)

[2.1. Introduction 5](#_Toc48567597)

[2.2. Road Safety and Accident 5](#_Toc48567598)

[2.3. Data Mining 9](#_Toc48567599)

[2.4. Predictive Analytics 13](#_Toc48567600)

[2.5. Other Relevant Articles 17](#_Toc48567601)

[Chapter 3 – Research Methodology 18](#_Toc48567602)

[3.1. Introduction 18](#_Toc48567603)

[3.2. The Research Onion 18](#_Toc48567604)

[3.3. Tools Used for the Analysis 20](#_Toc48567605)

[3.4. Implementation Methodology 20](#_Toc48567606)

[3.5. Experiment and Objectives 23](#_Toc48567607)

[3.6. Data Understanding 24](#_Toc48567608)

[3.6.1. Data Acquisition 24](#_Toc48567609)

[3.6.2. Data Description 24](#_Toc48567610)

[3.6.3. Data Exploration 27](#_Toc48567611)

[3.7. Data Preparation 27](#_Toc48567612)

[3.7.1. Data Cleaning 27](#_Toc48567613)

[3.7.2. Data Merging 28](#_Toc48567614)

[3.7.3. Data Transformation 29](#_Toc48567615)

[3.7.4. Data Construction 29](#_Toc48567616)

[3.8. Modelling 30](#_Toc48567617)

[3.8.1. Modelling Technique Selection 30](#_Toc48567618)

[3.8.2. Model Building 31](#_Toc48567619)

[3.9. Model Evaluation 32](#_Toc48567620)

[3.10. Limitations of the methodology 34](#_Toc48567621)

[Chapter 4 – Findings and Analysis 35](#_Toc48567622)

[4.1. Introduction 35](#_Toc48567623)

[4.2. Data Visualization and Analysis 35](#_Toc48567624)

[4.2.1. Accident Severity 35](#_Toc48567625)

[4.2.2. Day of the Week 36](#_Toc48567626)

[4.2.3. Casualty Gender 37](#_Toc48567627)

[4.2.4. Light Conditions 38](#_Toc48567628)

[4.2.5. Weather Conditions 40](#_Toc48567629)

[4.2.6. Road Surface Conditions 42](#_Toc48567630)

[4.2.7. Special Conditions at Site 44](#_Toc48567631)

[4.2.8. Month 46](#_Toc48567632)

[4.2.9. Hour 47](#_Toc48567633)

[4.2.10. Top 5 Districts and Highways 49](#_Toc48567634)

[4.2.11. Propulsion Type 51](#_Toc48567635)

[4.2.12. Casualty Class 52](#_Toc48567636)

[4.2.13. Age Band of Casualties 52](#_Toc48567637)

[4.3. Predictive Modelling and Analysis 53](#_Toc48567638)

[4.3.1. Prediction for Accident Severity 54](#_Toc48567639)

[4.3.2. Prediction for number of casualties 56](#_Toc48567640)

[4.3.3. Prediction for Number of Vehicles 62](#_Toc48567641)

[4.3.4. Prediction for Gender of Casualties 67](#_Toc48567642)

[4.3.5. Prediction for Average Age of the Casualties 70](#_Toc48567643)

[Chapter 5 – Discussion and Conclusions 76](#_Toc48567644)

[5.1. Introduction 76](#_Toc48567645)

[5.2. Discussion 76](#_Toc48567646)

[5.3. Conclusion and Reflection 78](#_Toc48567647)

[References 79](#_Toc48567648)

[Glossary of terms 82](#_Toc48567649)

[List of Abbreviations 83](#_Toc48567650)

[Appendix 84](#_Toc48567651)

##### List of Tables

[Table 3.6.2.1 : Description of the required features present in the data 26](#_Toc48568798)

[Table 4.3.1.1 : Count of training samples of each severity class for different class imbalance handling techniques 55](#_Toc48568799)

[Table 4.3.1.2 : Summary of machine learning models for prediction of accident severity 56](#_Toc48568800)

[Table 4.3.2.1 : VIF value of the columns for the prediction of number of casualties 58](#_Toc48568801)

[Table 4.3.2.2 : Descriptive statistics of the number of casualties 59](#_Toc48568802)

[Table 4.3.2.3 : RMSE for different values of k for prediction of number of casualties 60](#_Toc48568803)

[Table 4.3.2.4 : Comparison of different models for the prediction of number of casualties 60](#_Toc48568804)

[Table 4.3.3.1 : VIF value of the columns for the prediction of number of vehicles 63](#_Toc48568805)

[Table 4.3.3.2 : Descriptive statistics of the number of vehicles 65](#_Toc48568806)

[Table 4.3.3.3 : RMSE for different values of k for prediction of number of vehicles 65](#_Toc48568807)

[Table 4.3.3.4 : Comparison of different models for the prediction of number of vehicles 65](#_Toc48568808)

[Table 4.3.4.1 : Count of training samples of each gender class for different class imbalance handling techniques 68](#_Toc48568809)

[Table 4.3.4.2 : Summary of machine learning models for prediction of gender of casualties 69](#_Toc48568810)

[Table 4.3.5.1 : VIF value of the columns for the prediction of average casualty age 72](#_Toc48568811)

[Table 4.3.5.2 : Descriptive statistics of the average casualty age 73](#_Toc48568812)

[Table 4.3.5.3 : RMSE for different values of k for prediction of average casualty age 73](#_Toc48568813)

[Table 4.3.5.4 : Comparison of different models for the prediction of average casualty age 74](#_Toc48568814)

##### List of Figures

[Figure 1.1 .1: CRISP-DM model (Smart Vision - Europe, n.d.) 2](#_Toc48151807)

[Figure 1.2 .1: Leading causes of death, all ages, 2016 (World Health Organization, 2018) 3](#_Toc48151808)

[Figure 2.2.1 : List of variables in STATS19 dataset (Department of Transport UK, 2019) 7](#_Toc48151809)

[Figure 2.2.2 : The conceptual relationship between the economy and road safety (OECD/ITF, 2015) 8](#_Toc48151810)

[Figure 2.2.3 : Impact of unemployment on road safety (OECD/ITF, 2015) 8](#_Toc48151811)

[Figure 2.3.1 : The data processing pipeline for the data mining process (Aggarwal, 2015) 10](#_Toc48151812)

[Figure 2.3.2 : CRISP-DM reference model (Pete et al., 2000) 11](#_Toc48151813)

[Figure 2.3.3 : Typical corrupted variation of CRISP-DM (Taylor, 2017) 13](#_Toc48151814)

[Figure 2.4.1 : Predictive analytics value chain (PAT Research, n.d.) 14](#_Toc48151815)

[Figure 2.4.2 : Example of a decision tree (Scikit Learn, n.d.) 15](#_Toc48151816)

[Figure 2.4.3 : Prediction rule for the Naïve Bayes algorithm (Scikit Learn, n.d.) 15](#_Toc48151817)

[Figure 3.2.1: The research ‘onion’ (Saunders, Lewis and Thornhill, 2007) 18](#_Toc48151818)

[Figure 3.4.1: Experiment methodology for the project 21](#_Toc48151819)

[Figure 3.7.1.1: Count of missing data in the accident DataFrame 28](#_Toc48151820)

[Figure 3.9.1: Precision formula (Shung, 2018) 32](#_Toc48151821)

[Figure 3.9.2: Recall formula (Shung, 2018) 33](#_Toc48151822)

[Figure 3.9.3: F1-score formula (Shung, 2018) 33](#_Toc48151823)

[Figure 4.2.1.1: Distribution of accidents for each severity class 36](#_Toc48151824)

[Figure 4.2.2.1: Count of accidents for each day of the week 37](#_Toc48151825)

[Figure 4.2.3.1: Distribution of casualties for each gender class 37](#_Toc48151826)

[Figure 4.2.4.1: Total count of accidents for each light condition 38](#_Toc48151827)

[Figure 4.2.4.2: Count of fatal accidents for each light condition 39](#_Toc48151828)

[Figure 4.2.4.3: Proportion of accidents that were fatal under each light condition 39](#_Toc48151829)

[Figure 4.2.5.1: Total count of accidents for each weather condition 40](#_Toc48151830)

[Figure 4.2.5.2: Count of fatal accidents for each weather condition 41](#_Toc48151831)

[Figure 4.2.5.3: Proportion of accidents that were fatal under each weather condition 41](#_Toc48151832)

[Figure 4.2.6.1: Total count of accidents for each road surface condition 42](#_Toc48151833)

[Figure 4.2.6.2: Count of fatal accidents for each road surface condition 43](#_Toc48151834)

[Figure 4.2.6.3: Proportion of accidents that were fatal under each road surface condition 43](#_Toc48151835)

[Figure 4.2.7.1: Total count of accidents for each special condition 44](#_Toc48151836)

[Figure 4.2.7.2: Count of fatal accidents for each special condition 45](#_Toc48151837)

[Figure 4.2.7.3: Proportion of accidents that were fatal under each special condition 45](#_Toc48151838)

[Figure 4.2.8.1: Total count of accidents for each month 46](#_Toc48151839)

[Figure 4.2.8.2: Count of fatal accidents for each month 46](#_Toc48151840)

[Figure 4.2.8.3: Proportion of accidents that were fatal under each month 47](#_Toc48151841)

[Figure 4.2.9.1: Total count of accidents for each hour 47](#_Toc48151842)

[Figure 4.2.9.2: Count of fatal accidents for each hour 48](#_Toc48151843)

[Figure 4.2.9.3: Proportion of accidents that were fatal under each hour 48](#_Toc48151844)

[Figure 4.2.10.1: Accident hotspots in Great Britain 49](#_Toc48151845)

[Figure 4.2.10.2: Top 5 accident-prone district local authorities 50](#_Toc48151846)

[Figure 4.2.10.3: Top 5 accident-prone highway local authorities 50](#_Toc48151847)

[Figure 4.2.11.1: Count of accidents for each propulsion type 51](#_Toc48151848)

[Figure 4.2.11.2: Average age of vehicles with petrol propulsion in top 5 districts 51](#_Toc48151849)

[Figure 4.2.12.1: Distribution of casualties among different casualty classes 52](#_Toc48151850)

[Figure 4.2.13.1: Count of casualties in each age band 53](#_Toc48151851)

[Figure 4.3.1: Sequence of predictive objectives 53](#_Toc48151852)

[Figure 4.3.1.1: Input variables for the prediction of accident severity 54](#_Toc48151853)

[Figure 4.3.2.1: Initial set of input variables for the prediction of number of casualties 57](#_Toc48151854)

[Figure 4.3.2.2: Final set of input variables for the prediction of number of casualties 58](#_Toc48151855)

[Figure 4.3.2.3: Distribution plot of number of casualties 59](#_Toc48151856)

[Figure 4.3.2.4: Q-Q plot of residuals for the prediction of number of casualties 61](#_Toc48151857)

[Figure 4.3.2.5: Distribution of residuals for the prediction of number of casualties 61](#_Toc48151858)

[Figure 4.3.2.6: Fitted vs residuals plot for the prediction of number of casualties 62](#_Toc48151859)

[Figure 4.3.3.1: Initial set of input variables for the prediction of number of vehicles 62](#_Toc48151860)

[Figure 4.3.3.2: Final set of input variables for the prediction of number of vehicles 64](#_Toc48151861)

[Figure 4.3.3.3: Distribution plot of number of vehicles 64](#_Toc48151862)

[Figure 4.3.3.4: Q-Q plot of residuals for the prediction of number of vehicles 66](#_Toc48151863)

[Figure 4.3.3.5: Distribution of residuals for the prediction of number of vehicles 66](#_Toc48151864)

[Figure 4.3.3.6: Fitted vs residuals plot for the prediction of number of vehicles 67](#_Toc48151865)

[Figure 4.3.4.1: Input variables for the prediction of the gender of casualties 67](#_Toc48151866)

[Figure 4.3.4.2: Distribution of accidents for each gender class 68](#_Toc48151867)

[Figure 4.3.5.1: Initial set of input variables for the prediction of average age of casualties 70](#_Toc48151868)

[Figure 4.3.5.2: Final set of input variables for the prediction of average casualty age 72](#_Toc48151869)

[Figure 4.3.5.3: Distribution plot of the average casualty age 72](#_Toc48151870)

[Figure 4.3.5.4: Q-Q plot of residuals for the prediction of average casualty age 74](#_Toc48151871)

[Figure 4.3.5.5: Distribution of residuals for the prediction of average casualty age 75](#_Toc48151872)

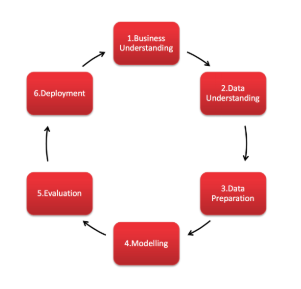
[Figure 4.3.5.6: Fitted vs residuals plot for the prediction of average casualty age 75](#_Toc48151873)

# Chapter 1 - Introduction

## Introduction

Road traffic accidents are among the leading cause of deaths in the world. According to the World Health Organization (2020), approximately 1.35 million people lose their lives each year due to road traffic accidents across the globe. In Great Britain, 1,784 road deaths were reported in 2018 by the department of transport which was on par with 1,754 deaths in 2012 suggesting that there has not been any significant improvement in 6 years (Robineau, 2019). The key factor in 95% of all road accidents in Great Britain is a human error such as the influence of alcohol, driving inexperience, tiredness or illness (ROSPA, 2017). However, many other circumstances can impact the severity of a road accident such as lighting condition, weather condition, road surface, etc. It is also crucial to recognize the hotspots for accidents namely the districts and the highways where the majority of the accidents takes place. Identification of these elements that leads to fatalities on the road is an important aspect of road safety analysis which can be used to spread awareness among the commuters and the pedestrians towards safer roads.

This research project explores the literature related to accidents, road safety, data mining, data preparation and predictive machine learning algorithms. Classification algorithms such as Decision Tree Classifier, Random Forest Classifier and Gaussian Naïve Bayes are examined to predict the severity of future accidents and the gender of the casualties whereas, regression algorithms such as Linear Regression, Decision Tree Regressor and K-Nearest Neighbour are also studied to predict the number of casualties, the number of vehicles and the average age for the casualties involved in future accidents. The models are evaluated and compared to identify a suitable machine learning algorithm for each task. The experiments carried out for this research project are designed as per the standard CRISP-DM methodology of understanding and preparing the data, selection and implementation of appropriate modelling technique for the given task followed by the model evaluation and outlining of key observations as shown in Figure 1.1.1 (Smart Vision - Europe, n.d.).

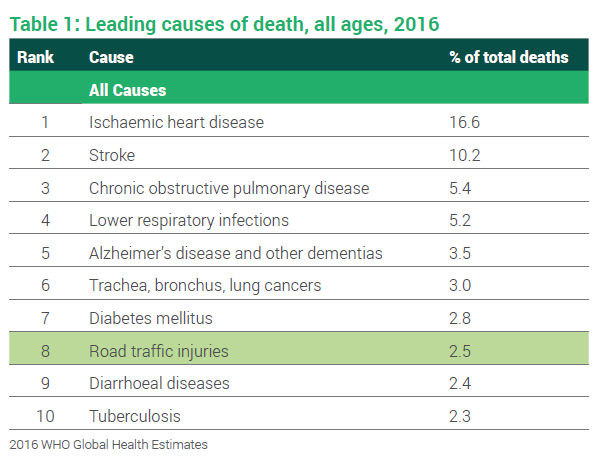


1. .1: CRISP-DM model (Smart Vision - Europe, n.d.)

The introduction is followed by the literature review section. In that chapter, literature related to road safety and accidents will be discussed thoroughly to get a better understanding of the need for this research. The process of data mining and CRISP-DM model will also be explored which is a crucial activity in any data analytics project. Apart from this, literature regarding predictive analytics and machine learning aspects of classification and regression along with a discussion on some relevant articles will also be touched upon in that chapter. The research methodology chapter will showcase the techniques used during the research project and the justification for their use. The effectiveness of the research methods regarding the proposed objectives will be assessed as well as the limitations of the data and research methods will also be reviewed in that chapter. The next chapter will be the findings and analysis where the results of the data analysis will be discussed, and the factors present in the majority of the road accidents will be identified. All the other significant observations will also be presented in that chapter. The predictive models will be evaluated in that section and their usefulness will be considered. The discussion and conclusion chapter will further investigate the findings of the project to answer the research objectives. That chapter will also summarize the entire research and present a comparison between this research and the previous works related to the topic. The scope of future work in the area of road safety will also be addressed in that chapter.

## Background

This chapter focuses on the concept of the road traffic accident and the role of road safety in the prevention of accidents. Deaths and injuries caused by road traffic accidents are a major concern all around the world and current patterns indicate that the situation will remain the same within a reasonable time-frame (World Health Organization, 2018). In fact, it is the leading cause of death for children and young adults aged 5-29 years (World Health Organization, 2020). As per WHO’s Global Status Report on road safety 2018, road traffic injuries were the eighth highest cause of death among people of all ages in 2016 (World Health Organization, 2018).



1. .1: Leading causes of death, all ages, 2016 (World Health Organization, 2018)

There are several factors which impact the likelihood of injury or death on roads such as the amount of traffic, the type of travel, the type of vehicle, the conditions on the road and the behaviour of fellow drivers (Rosenfeld, 2017). Rosenfeld (2017) suggested that infrastructural improvements like installation of rumble strips, median barriers, roundabouts, turning lanes and pedestrian refuge areas can be deployed to decrease roadway fatalities. Apart from infrastructural modifications on roads, several other strategies such as the reduction in the speed limit, advancements in safety equipment within the vehicle, use of child safety seats, stricter law enforcement and the prohibition on the use of electronic devices while driving may also help in making the roads safer. To improve the safety on roads, factors which leads to accidents on roads needs to be identified so that appropriate countermeasure can be introduced. Road safety strategies must be based on rigorous analysis and interpretation of historical data along with the consideration of implementation costs and side effects (Wegman *et al.*, 2015).

## Research Aim and Objectives

The main research question for this project is to identify whether the historical data related to road safety and accidents for Great Britain help in predicting the likelihood of the severity of an accident and the details about the casualties for a specific location with particular circumstances?

The objectives of the research are:

* To analyse the historical data and study the impact of various conditions such as weather, lighting, road surface and any other special condition on the severity of the accident to understand the need of any infrastructural changes and spread awareness in areas more prone to accidents
* To visualize the accident hotspots on the map of Great Britain and graphically represent various statistics regarding the accidents and casualties
* To predict the details about future accidents including the severity, the number of casualties and vehicles, the gender of the casualties and the average age of the casualties
* To implement machine learning algorithms related to classification and regression for predictive analytics domain of the project
* To suggest future work which can be carried out based on the findings of this research project

# Chapter 2 – Literature Review

## 2.1. Introduction

This chapter discusses current literature related to road safety, traffic accidents, data mining and predictive machine learning algorithms relevant to this research project. The CRISP-DM methodology, which is a standard data mining workflow, is also explored in this chapter. This chapter also explores the predictive analytics and literature regarding classification and regression algorithms that were used in this project to meet the research objectives. Since accident analysis is crucial in maintaining road safety, some work has already been done in this area which includes classification of road accident hotspots in London, prediction of expected accidents on the highways in California and analysis on the impact of speed limit violation, alcohol consumption limit violation and failure to wear a seat belt on accidents in the United States. An overview of road safety and accidents is presented to outline the comprehension of the research topic. The overview is followed by a short analysis of the role of UK road safety data, accident statistics, data mining and predictive analytics for prediction.

## 2.2. Road Safety and Accident

A road traffic accident can be defined as follows:

“An accident which occurred or originated on a way or street open to public traffic; resulted in one or more persons being killed or injured, and at least one moving vehicle was involved.” (OECD Health Statistics, 2019).

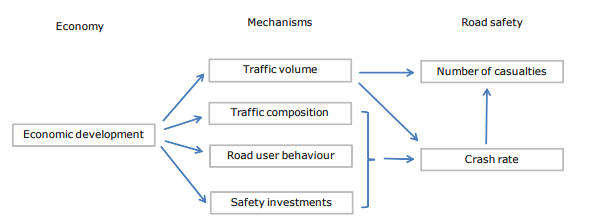
From this definition, it can be concluded that road accidents are a fatal threat to the people around the world. It was the eighth highest cause of death among people of all ages in 2016 (World Health Organization, 2018). According to the road safety factsheet issued by The Royal Society for the Prevention of Accidents (ROSPA) in 2017, car occupants constitute to 46% of the total fatalities on the roads in Great Britain which comprise 68% drivers and 32% passengers. Young drivers aged 17-24 years were involved in 20% of fatal or serious crashes in 2016 despite being only 7% of UK full driving license holders (ROSPA, 2017). The factsheet also states that the most common factor associated with 95% of the total road accidents in Great Britain is human error which involves the influence of alcohol, lack of driving experience, tiredness, illness, etc. Hence, the young drivers should be extra cautious on the roads and ensure that they are familiar with the road safety guidelines before heading out. The factsheet reveals that the road environment, which incorporated road design and the road surface, is responsible for approximately 12% of the total accidents on the road. The progress towards achieving safer roads can be accelerated with the help of an integrated approach involving enhanced safety standards for roads and vehicles, legislation against risky practices such as speeding, as well as safeguarding prompt access to emergency care (World Health Organization, 2018). This approach relies heavily on the availability of comprehensive and reliable data on road traffic crashes, injuries and deaths to monitor the progress (World Health Organization, 2018). Thus, the availability of trustworthy and up-to-date data concerning the conditions during road accidents and the resulting injuries or casualties is of paramount significance for the creation and regulation of accident preventive measures. Reliable data is also essential as justification for the introduction of new safety measures or the upgradation of existing ones.

In Great Britain, the accident statistics are collected by police officers. These data are gathered through attending the scene of the accident or from public reporting of the accident in the police station after the incident using the STATS19 reporting system (Robineau, 2019). In the 2018 annual report on reported road casualties in Great Britain issued by the Department for Transport (DfT), Robineau (2019) highlights that since the people are not obliged to report all the accidents and personal injuries to the police, the data does not represent all the accidents or casualties in Great Britain. She also suggests that to obtain the absolute casualty count, data from other sources should also be considered like National Travel Survey, Hospital Episode Statistics, Compensation recovery unit data and Motor Insurance claims. Although it is not perfect, the STATS19 data is still the most detailed, complete and reliable single source of information about road accidents, casualties and involved vehicles (Robineau, 2019). The features recorded by the DfT for the accidents, casualties and vehicles associated with road crashes in Great Britain are listed in Figure 2.2.1.



1. : List of variables in STATS19 dataset (Department of Transport UK, 2019)

The International Transport Forum (ITF) at the Organisation for Economic Co-operation and Development (OECD) observed an interesting relationship between the economy and road safety that whenever economic growth declines, and particularly when unemployment increases, road safety improves (OECD/ITF, 2015). It defined economic development as the growth or decline of economic activity, measured by criteria like the volume growth of the Gross Domestic Product (GDP) or the unemployment rate. The ITF’s report in 2015 argues that economic development may impact the volume of traffic on roads, thereby increasing the risk of a crash. The report indicates that the economic development may also cause a rise in the share of young drivers; or trigger a behavioural change in traffic such as over speeding or drink-driving. However, a counter-argument was also highlighted in the report that economic development can also cause an increase in road safety by allowing the government, road users and companies to invest more in safety (OECD/ITF, 2015). Through the ITF’s article, it can be apprehended that economic development influences road safety by regulating the volume and nature of the road traffic. The growth in the economy enables more people to have vehicles and elevates the number of young drivers which can lead to increased cases of road crashes. On the flip side, a growing economy can also empower the government, vehicle manufacturers and other organizations to implement superior safety mechanisms to reduce road crashes. Figure 2.2.2 depicts the conceptual framework that was put forward by the ITF to explain the relationship between economy and road safety.

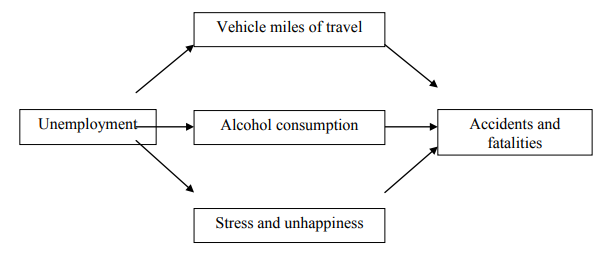


1. : The conceptual relationship between the economy and road safety (OECD/ITF, 2015)

Leigh and Waldon (1991) presented three hypotheses to establish the relationship between unemployment and road safety which are as follows:

* Increased unemployment is associated with a reduced amount of driving.
* Increased unemployment may influence alcohol consumption, but the direction of the influence is indeterminate.
* Increased unemployment may increase aggregate levels of stress and unhappiness; this would, in turn, be expected to increase the number of accidents.

A casual model can be made from these hypotheses as shown in Figure 2.2.3.

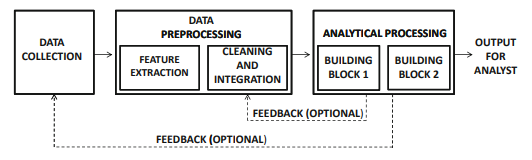


1. : Impact of unemployment on road safety (OECD/ITF, 2015)

However, there is no available data to support the claim that drivers are more stressed during economic downfall or their anxiety has a negative impact on their driving performance and safety (OECD/ITF, 2015). It suggests that the hypothesis is only speculation; which does not imply that it is false. This exemplifies a persistent issue with econometric models of road safety, that certain factors based on the behavioural adaptation of road users are not directly observable (OECD/ITF, 2015). Although each causal mechanism in the model cannot be supported with evidence, based on the statistics gathered from fourteen countries including Austria, Belgium, Denmark, Finland, France, Germany, Great Britain, Ireland, Japan, Netherlands, Norway, Sweden, Switzerland, and the United States, a decline in the number of traffic fatalities by about 4,850 was observed during the economic recession in 2009 and 2010 (OECD/ITF, 2015). Therefore, much like an improvement in economic development could lead to an increased number of road accidents, as mentioned earlier, a depreciation in the economy in the form of unemployment could lead to decreased number of road accidents by shrinking the traffic volume on roads. However, amplified negative behavioural impacts caused by unemployment could also escalate the cases of road accidents. Based on these arguments, it can be interpreted that the implementation of improved road safety measures, based on strong analysis and investigation of accident-related data, is a more reliable way of preventing accidents compared to dependence on economic development to do the same by decreasing the volume of traffic.

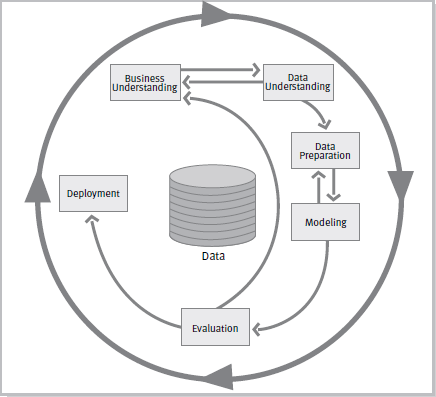
## 2.3. Data Mining

Data mining can be defined as the study of collecting, cleaning, processing, analysing, and gaining useful insights from data (Aggarwal, 2015). It refers to the process of converting raw data into useful information by identifying meaningful trends or patterns (Twin, 2019). It mainly involves the exploration and analysis of huge data blocks to extract hidden relationships amongst the data. These hidden relationships can help in the discovery of new solutions for a problem or the improvement of existing solutions. A wide range of data types such as relational databases, data warehouses, transactional databases, multimedia databases, time-series databases, etc., can be examined through data mining (Rajput, n.d.). The overall data mining process is illustrated in Figure 2.3.1. The various building blocks within the analytical processing block represents the design for the solution of a particular application (Aggarwal, 2015).



1. : The data processing pipeline for the data mining process (Aggarwal, 2015)

This literature review mainly focuses on those data mining techniques that are of relevance to the project, which falls under the CRISP-DM methodology. It stands for the Cross-Industry Standard Process for Data Mining and lays out a well-organised approach to the data mining process (Smart Vision - Europe, n.d.). It provides a non-proprietary and freely available standard process for fitting data mining into the general problem-solving strategy of a business or research unit (Larose, 2015). The steps involved in CRISP-DM can be performed in a different order based on the requirement. The processes involved in a CRISP-DM model is shown in Figure 2.3.2. The reference model in Figure 2.3.2 depicts the life cycle of a data mining project and it contains different phases of a project, their respective tasks and the relationship between these tasks (Pete *et al.*, 2000). Pete *et al*. (2000) also mentioned that depending on the goal, background and data; there could be relationships between any data mining tasks. CRISP-DM illustrates the natural workflow of a data mining task and it is often described as “the first step towards defining a data science methodology” (Saltz, 2015, p.1).



1. : CRISP-DM reference model (Pete et al., 2000)

The first stage in a CRISP-DM style model lays out the focus on the business understanding which is required to align technical work with business needs. It also prevents the Data Scientists to charge towards a problem without comprehending the business objectives (Data Science Project Management, n.d.). Once the problem is understood from a business perspective, the objectives are converted into a data mining problem (Pete et al., 2000).

The second stage is data understanding which includes the acquisition of data and exploration of data for identification of key attributes and the relationship between them (Smart Vision - Europe, n.d.). Pete et al. (2000) also pointed out the value of gaining familiarity with the data, identification of data quality problems and detection of interesting subsets from the data at this stage of the CRISP-DM model. It is vital at this stage to verify the quality of the acquired data and perform simple aggregations to summarize the data.

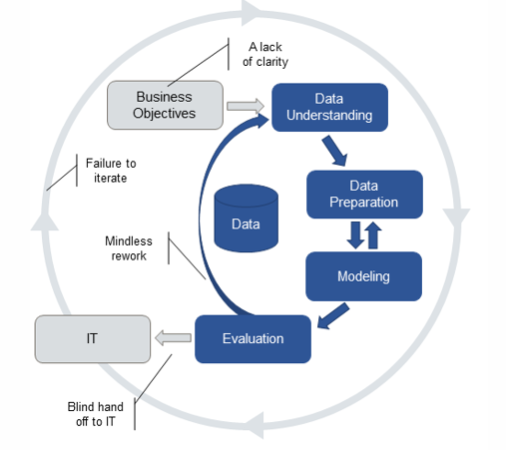
The third stage is data preparation which requires the cleansing of data, the transformation of data, the integration of data from multiple sources and the usage of feature engineering for the generation of new attributes for analysis (Manasson, 2019). All these activities lead to the creation of a final dataset from the raw data that can be fed to the machine learning models (Pete et al., 2000). Their journal article also mentioned that data preparation tasks may need to be performed multiple times without any prescribed order.

The fourth stage is modelling which incorporates the selection of modelling technique based on the target, generation of training and testing data, and execution of the model on training data (Smart Vision - Europe, n.d.). It should be noted that many modelling techniques can be applied to the same data mining problem type (Pete et al., 2000). They pointed out that since some techniques have specific requirements on the form of the data, it is often needed to go back to the data preparation phase before the model could be applied.

The fifth stage is the evaluation which involves the assessment of the model on testing data and its evaluation in terms of the business requirements. Pete et al. (2000) introduced an essential objective for this stage to identify important business issues that have not been properly considered in the chosen model(s). In this stage, the statistical significance of the findings should also be estimated for those projects which demand statistical analysis (Manasson, 2019).

The final stage of the CRISP-DM model entails the deployment of the selected model(s) as well as its monitoring and maintenance (Smart Vision - Europe, n.d.). The aim of the deployment strategy should be to organize and present the model in such a way that the customer could use it (Pete et al., 2000). They highlighted that based on the requirement, the deployment phase can be as simple as the generation of a report or as complex as an implementation of a repeatable data mining process across the enterprise. Manasson (2019) added that the deployment strategy should safeguard the readability, reproducibility and maintenance of the analysis. Thus, the CRISP-DM methodology lays out the groundwork for a data mining project and guides the analyst or researcher through all the essential processes required for the successful completion of the task and the fulfilment of the business objectives.

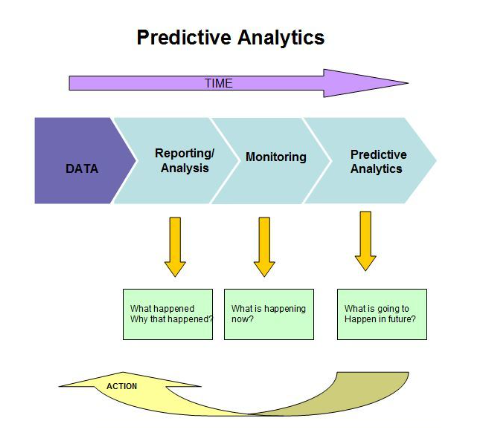
Although CRISP-DM is the standard for any data mining process, it has its weaknesses (Data Science Project Management, n.d.). It is a task-centric method and does not confronts team and communication challenges (Data Science Project Management, n.d.). Due to the lack of proper communication, the project team may end up with an unclear understanding of the business objectives which can result in the creation of interesting models that do not meet the actual business requirement. No matter how fancy the model is, if it does not solve the intended business problem, it is useless. The project team needs to keep the model updated and ensure that the model can adapt to the changes in business circumstances and data patterns (Taylor, 2017). Taylor also observed that sometimes project teams do not follow the CRISP-SM methodology in its entirety and adopt shortcuts which gives rise to a corrupted version of the model as shown in Figure 2.3.3. This approach introduces additional complications such as blind hand-off to IT for model deployment which can lead to unusable final product with no business impact (Taylor, 2017). The unnecessary reworks after model evaluation in this corrupted model can also be linked to the unclear understanding of business objectives at the beginning of the process flow.



1. : Typical corrupted variation of CRISP-DM (Taylor, 2017)

## 2.4. Predictive Analytics

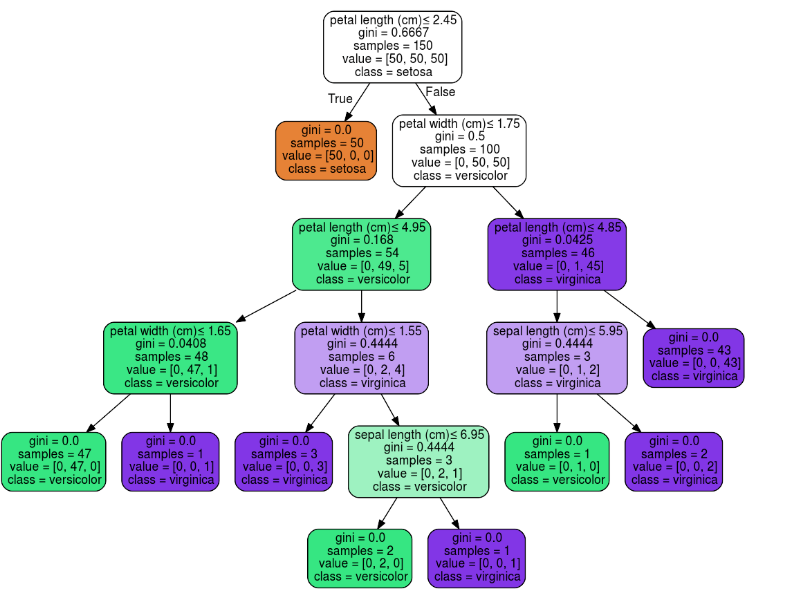
Predictive analytics can be defined as the process of extracting information from large data sets to make predictions and estimates about future outcomes (Larose, 2015). Predictive analytics is implemented to make predictions about future events by applying data mining, statistics, machine learning and artificial intelligence to the data (PAT Research, n.d.). Predictive power or predictive accuracy indicates the ability of a model to accurately predict the outcome of new observations (Shmueli and Koppius, 2011). It can be applied to identify risks and opportunities for the future through the patterns found in historical and transactional data (PAT Research, n.d.). Organizations can utilize predictive analytics to become proactive and anticipate outcomes and behaviours based on data instead of hunch (PAT Research, n.d.). It can be applied to a variety of problems such as predicting the price of ongoing eBay auctions, predicting the future sales of a movie in box office based on its online rating and predicting the repeat visits and likelihood of purchase of online customers (Shmueli and Koppius, 2011).



1. : Predictive analytics value chain (PAT Research, n.d.)

Predictive analytics and machine learning are often considered the same thing but there is a slight difference between the two of them. Predictive analytics is aimed towards the estimation of future outcomes with the help of statistics and analysis of historical data (Halton, 2019). Whereas machine learning enables computers to learn without being explicitly programmed (Samuel, 1969). Thus, it can be inferred that machine learning is a tool to perform predictive analytics by learning from a given set of input data and it does not need to be programmed explicitly for every possible test case. A predictive model is built after careful speculation of the problem and the target since the model type and methods rely on it (Shmueli and Koppius, 2011). Predicting an outcome value from a given new set of observations is a common objective in predictive analytics. Depending on the type of outcome, this goal can be termed as classification, for a categorical outcome; or regression, for a numerical outcome (Shmueli and Koppius, 2011) (Abrams, n.d.).

Classification techniques are supervised machine learning methods that allow the computer program to identify the class of a new observation based on the given historical data (Waseem, 2019). It is referred to as supervised learning since the algorithm uses an example data set to learn the structure of the groups (Aggarwal, 2015). Aggarwal (2015) mentioned that a typical classification algorithm has two phases: the training phase and the testing phase. In the training phase, the algorithm learns to create a mathematical model of the labelled groups in the training data set. In the testing phase, the model predicts the class label for one or more unseen test instances. Some of the widely used classification algorithms include Decision Tree Classifier, Random Forest Classifier, Naïve Bayes Classifier, Logistic Regression, Support Vector Machines and Neural Network (Aggarwal, 2015). In this project Decision Tree, Random Forest and Naïve Bayes were selected for the classification tasks. The reason for their selection will be explained in the next chapter. Tree-based models exercise simple if-else decision which is learned from the input data features to predict the value of a target variable. They also work well with most types of data and require little data preparation (Scikit Learn, n.d.).



1. : Example of a decision tree (Scikit Learn, n.d.)

Naïve Bayes algorithm works on the Bayes’ theorem and it is naïve in the sense that it assumes conditional independence between the variables (Scikit Learn, n.d.).



1. : Prediction rule for the Naïve Bayes algorithm (Scikit Learn, n.d.)

Regression techniques are aimed at predicting a target value based on independent predictors (Gandhi, 2018). It is a process of mathematically determining those input variables that have an impact on the target variable (Gallo, 2015). The key variable that the regression algorithm aims to predict is also called the dependent variable and the variables that are suspected to have an impact on the dependent variable are called independent variables. Gallo (2015) emphasises that while working with the regression or any other technique that attempts to understand the relationship between the variables, it is important to remember that correlation does not always indicate causation. Hence, it can be understood that a good understanding of the input variables and domain expertise is needed to determine whether high correlation refers to causation or not. He also emphasised that before implementing any machine learning algorithm, it is crucial to re-evaluate whether the results fit the understanding of the situation. Some of the widely used regression algorithms for prediction of a continuous target variable are linear regression, polynomial regression, stepwise regression, ridge regression and lasso regression (Ray, 2015).

This research project often deals with classification tasks involving more than two classes, such classification problems are called multiclass classification (Nabi, 2018). Another important aspect of the data under consideration is the difference in the number of samples for each class. The number of data samples that belong to the accident severity class ‘Fatal’ constitutes only 1.3% of the entire data, whereas the number of data samples that belong to accident severity class ‘Slight’ constitutes 82.2% of the entire data. Such datasets where the classes are not represented equally are called imbalanced datasets (Chawla *et al.*, 2002). Training a predictive machine learning algorithm to correctly perform classification tasks on an imbalanced dataset is difficult since there is a high probability that the algorithm will get over-fit and heavily biased towards the class with the majority of the samples (Nabi, 2018). To counter this complication, one can reduce the count of training samples of the majority class or increase the count of the training samples of the minority class. The process of removing samples of the majority class is called *undersampling* and the process of increasing the samples of minority class is called *oversampling* (Brownlee, 2020). Removal of majority class samples is straightforward, but the increment of minority class samples can be done in different ways. One way to achieve this is through random replication of observations from the minority class. This process is known as random oversampling. Another approach to the over-sampling process is through the creation of ‘synthetic’ examples of the minority class based on the existing observations. This over-sampling algorithm was developed by Chawla *et al.* in 2002 and it was named ‘Synthetic Minority Oversampling Technique’ or SMOTE. Based on tests conducted using several imbalanced datasets, it was observed that the performance of SMOTE was better than random under-sampling and random over-sampling (Chawla *et al.*, 2002). The issue of class imbalance for the classification tasks in this research was dealt with the help of the SMOTE algorithm.

## 2.5. Other Relevant Articles

This paragraph highlights the ideas and techniques mentioned in a few journal articles and dissertations that discussed road safety, accidents and predictive algorithms in that domain. A general copula-based multivariate count regression model with correlated random effects within a Bayesian framework could be used to assess the effects of roadway characteristics or environmental factors on crash counts by severity level or by collision type (Park *et al.*, 2020). The research studied the dependence of multivariate random effects on the multivariate crash counts in California through simulation using copulas. High-resolution crash severity models based on driver injury severity can be developed using the Abbreviated Injury Scale (AIS) by body region (Kabli, Bhowmik and Eluru, 2020). Their research identified that age plays a vital role in determining injury severity and newer vehicles reduce the likelihood of a serious injury. C5.0, Chaid decision trees and Bayes net classification models can extract fatal accident prediction from UK STATS19 accident data (Connor, 2015). The aim of Connor’s dissertation was similar to one of the objectives of this research project. The experimentation carried out in his work classified accidents as fatal or non-fatal with the help of various machine learning models, however, the accuracy and precision performances were poor in many cases. A methodology using Geographical Information Systems (GIS) and Kernel Density Estimation can be used to study the spatial patterns of injury-related road accidents in London, UK (Anderson, 2009). A clustering methodology using environmental data and results from the first section was also presented in that journal article in order to create a classification of road accident hotspots. The findings of that research highlighted the Westminster district of Central London as there were many spatially extensive accidents due to large traffic flow and big coverage area of the hotspot. A Willingness-To-Pay (WTP) survey carried out in the UK provided information for the value of avoidance of serious injuries (O’Reilly *et al.*, 1994). The WTP approach derived monetary values of safety for the public who were affected by safety investment decisions. Thus, determining the maximum amounts that affected individuals would be willing to pay for improvement in their own or other’s safety. The survey results suggested that the accident history of a person did not have any significant impact on the valuation and the WTP amount depended on individual personality and psychological disposition, rather than the demographic factor or household income.

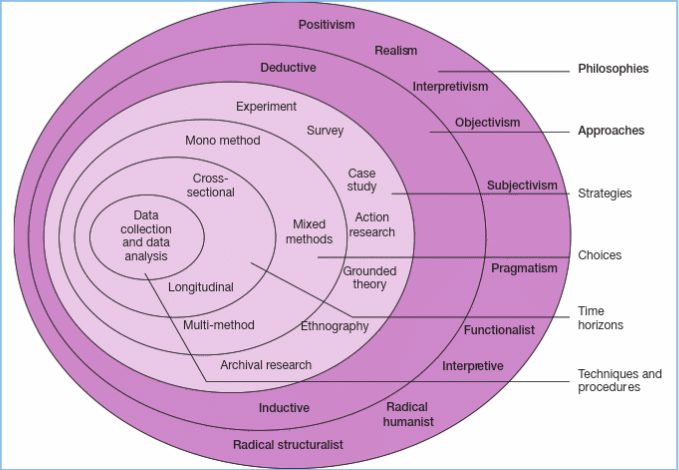
# Chapter 3 – Research Methodology

## 3.1. Introduction

This chapter discusses the methods adopted during the completion of the research project and scrutinizes them in relation to the extent to which they answer the research objectives. The procedures undertaken to identify the factors related to most of the accidents and to recognize the relationship between the accidents and the variables are explored in this chapter. The data used for the experiments along with the data preparation procedures employed to construct the data is also presented. In addition to that, the shortcomings of the data and the techniques used to deal with them are also discussed. This chapter discusses the predictive analytics techniques and different machine learning algorithms that were utilized to make predictions about the severity of future accidents, the number of casualties, and the count of vehicles involved in future accidents. Along with that, the algorithms for the prediction of gender of the majority of the casualties and the average age of the casualties are also analysed in this chapter. The reasons for the selection of an appropriate model for each objective will be provided and their limitations will also be discussed in this chapter.

## 3.2. The Research Onion

The design of a piece of research can be depicted with the help of the ‘research onion’ as shown in Figure 3.2.1. The research onion describes the steps needed for the development of an effective research methodology.



1. The research ‘onion’ (Saunders, Lewis and Thornhill, 2007)

Stainton (2020) provided a simple explanation for each layer of the research onion. The first layer of the research onion describes the philosophies associated with research and generally debates over the use of existing data or the actual reality which may not be evident in the existing data. Their understanding helps to create a practical approach to the research. For this research, historical data was used to study the impact of various factors on road accidents. The second layer of the research onion describes the approach in terms of being deductive or inductive. A deductive approach starts with the development of a theory and leads to the creation of strategies to test the theory. An inductive approach starts with the data and results in the creation of a theory based on the analysis of that data. This project adopted the deductive approach and tested the impact of environmental and infrastructural factors on the accidents. The third layer of the research onion focuses on the strategies to be adopted for the collection of data for the research. The strategy for this research project was based on archival research since it makes the use of existing information for the analysis. The fourth layer of the research onion presents the choice of method which can be mono, mixed or multi. This project implemented the mono method as it relied on the quantitative analysis of the past data. The fifth layer of the research onion touched upon the time horizon for the research. Since this research made use of pre-recorded data and had time constraints on its completion, it followed a cross-sectional method. The sixth layer of the research onion mentions the techniques for data collection and analysis. The data used for this research project was collected by the police force in Great Britain and was made publicly available at the UK government’s data repository.

The general plan to discover answers for the research question(s) is known as the research design (Saunders, Lewis and Thornhill, 2007). The research design should reflect the thoughts behind the selection of a particular research design. A research design is comprised of one or a combination of the following studies:

* Exploratory Study
* Descriptive Study
* Explanatory Study

An exploratory study is used to identify the reasons behind a phenomenon. It is employed to clarify the understanding of the problem and to uncover its precise nature. A descriptive study is aimed at the accurate portrayal of an event or situation. However, descriptive study in itself does not help in drawing conclusions from the data and other higher-order skills such as data evaluation skills and predictive skills are also needed in a project. Saunders, Lewis, and Thornhill (2007) described the descriptive study as “a means to end rather than an end in itself.” An explanatory study establishes a causal relationship between the variables. Statistical tests such as correlation can be applied to clearly view the relationship.

This research project incorporates all the above-mentioned studies. The exploratory study was exercised to gain insights about the research topic as discussed in chapter 2. The descriptive study was utilized to understand the variables within the dataset. For example, the proportion of accidents within each severity class was computed to check the volume of data for each of them. The explanatory study was used to find out the most impactful factors in a road accident. For example, the count of accidents under different road and weather conditions was calculated and visualized to detect the most influential category for each condition.

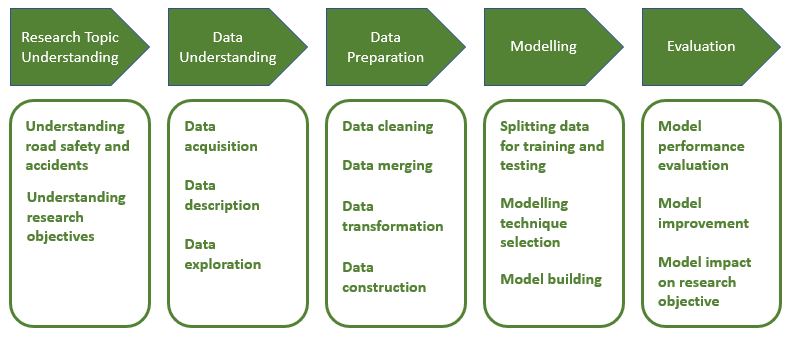
## 3.3. Tools Used for the Analysis

The methodology was executed with the help of the Python programming language. All the steps from reading the data to its analysis and implementation of predictive machine learning models were carried out in Jupyter notebook. The reason for the selection of Jupyter notebook for the project was that it enables the users to run and visualize the codes line-by-line as they progress with their project instead of compiling everything at once. Jupyter notebook is available with the Anaconda distribution of Python which has a cloud-based repository of more than 7,500 data science and machine learning packages (Anaconda, n.d.). Due to these reasons, it is a well-known Python IDE in the data science community. Tableau version 2019.2 was also used for the visualization of accident hotspots on the map of Great Britain.

The Python libraries used for data understanding and data preparation were mainly Numpy and Pandas. Matplotlib, Seaborn and Plotly libraries were employed for data visualization. Scikit-Learn (sklearn) library was extensively used for the predictive analytics aspect of the project. All the machine learning models, and their evaluation metric was made available in that package. Along with that, the Imbalanced Learning (imblearn) library, which contains the functionality for SMOTE and Random Under Sampler, was utilized in the classification tasks to deal with the class imbalance in the data.

## 3.4. Implementation Methodology

The methodology undertaken for the research and experimentation in this project was based on the Cross-Industry Standard Process for Data Mining (CRISP-DM) model. The guidance from the CRISP-DM model was taken to design the overall data life cycle for this project. However, the phases of the data life cycle from the CRISP-DM model were modified in accordance with the project objectives. The overview of all the phases of the data life cycle for the project is depicted in Figure 3.4.1.



1. Experiment methodology for the project

The primary stage of the methodology was to establish an understanding of the term road safety and road traffic accident. Each research objective was clearly understood and motivation behind them was recognized. This was needed to ensure that the research suitably recognised the problems with the existing road safety measures. It also helped with the selection of necessary features from the dataset for various predictive models used in the project.

The data understanding stage involved the identification of possible data sources and the acquisition of data from a trusted source. The data related to accidents, casualties and vehicles were available as Comma Separated Values (CSV) files for each year on the official UK government data repository (Department of Transport UK, 2019) which is also known as STATS19 dataset. The data from the year 2015 to 2018 was gathered and merged using the Python programming language to create a single DataFrame for each category i.e., accident, casualty and vehicle. The data was then explored to view all the features recorded for the accidents, casualties and vehicles related to road crashes in Great Britain. The casualty and vehicle DataFrames could be linked to accident DataFrame through a unique accident index for each accident. The descriptive statistics for each numerical column were studied to obtain the overview. Data visualization was employed to study the impact of various conditions such as lighting, weather, road surface or any other special condition on the accidents. The most affected districts and highways were also discovered. From the casualty’s aspect, the count of casualties in each age band and the proportion of casualties for each casualty class, like the driver, passenger or pedestrian, was also observed. The number of casualties based on their gender was calculated to understand their distribution for predictive models.

The next stage of the methodology was data preparation. In this stage, the count of missing values in each column was calculated. Since the volume of data was quite substantial, the missing values were removed without losing valuable information. The column for the time of the accident contained the hour and the minute for the accident which was later split into two separate columns for the hour and minute respectively for further analysis. The casualty DataFrame stored the details for the victims of the accidents. The most common gender of casualties for each accident and the average age of casualties for each accident from the casualties DataFrame was determined and merged with the accident DataFrame to create two new columns of casualty gender and average casualty age within the accident DataFrame for future predictions.

The modelling stage involved the creation of machine learning models for prediction of accident severity, number of casualties, number of vehicles, the gender of casualties and the average age of casualties for future accidents. The prediction for accident severity and casualty gender belonged to the classification aspect of predictive analytics and the prediction for the number of casualties, vehicles and the average age of casualties belonged to the regression aspect of predictive analytics. The algorithms tested for the classification tasks were: Decision Tree Classifier, Random Forest Classifier and Naïve Bayes Classifier; whereas algorithms tested for the regression tasks were: Linear Regressor, Random Forest Regressor and K Nearest Neighbour Regressor. The rationale behind the preference of tree-based algorithms for the predictive analytics area of this project was the excessive presence of label encoded columns in the data. To make use of distance-based algorithms, these label encoded columns need to be converted to dummy encoded columns or one-hot encoded columns so that the model does not get biased towards higher valued data points. The use of dummy encoding or one-hot encoding significantly increases the number of columns in the data; hence their use was avoided wherever possible. The next step in this stage was the division of data into two sets: one for training the model and the other for testing the model. For each target (dependent) variable of prediction tasks, the appropriate set of dependent variables from the data was selected and split into training and testing set. The models were then trained on the training set and predictions for the testing set were made.

The models were then evaluated based on their performance on the test set. For classification tasks, the accuracy of the model was checked, and the confusion matrix and classification report were examined to judge the model’s performance. For regression tasks, the root mean squared error (RMSE) and the R2 value were considered for the evaluation of the model’s performance. To improve the performance of regression models, few input variables were removed from the training and testing set upon inspection of multicollinearity indicated by the variation inflation factor (VIF). While selecting the appropriate model for each task, special care was taken to not judge the models solely on their quantitative summary but also on the extent to which they meet the solution for the intended problem. For example, during the selection of a final model for prediction of accident severity, apart from the consideration of the model’s overall accuracy, the model’s performance in the identification of individual severity classes was also investigated.

## 3.5. Experiment and Objectives

It was identified in chapter 2 that road accidents are one of the leading causes of deaths in the world. Therefore, the analysis of existing road safety data for the identification of key contributing factors is necessary to reduce the number of accidents in the future. The role of predictive analytics in predicting the severity of future accidents and the details about the associated casualties will help in providing prompt emergency services to the victims which could reduce the fatality count. Along with the analysis of the existing road safety data, this research project was also focused on the design of predictive models based on the STATS19 road safety dataset for accidents in Great Britain.

The first predictive objective of the project was the prediction of the severity of future accidents and classify them as fatal, serious or slight accidents represented by the numbers 1, 2 and 3 respectively. The second predictive objective of the project was the prediction of the number of casualties for a future accident. The third predictive objective of the project was the prediction of the number of vehicles associated with a future accident. The fourth predictive objective of the project was the prediction of the gender of the majority of the casualties in a future accident and classify them as majority male, majority female or both genders. The fifth predictive objective of the project was the prediction of the average age of the casualties in a future accident.

## 3.6. Data Understanding

The data understanding phase of the research methodology involved familiarisation with the dataset under consideration. It comprised of following steps:

* Data Acquisition
* Data Description
* Data Exploration

### 3.6.1. Data Acquisition

The data related to road accidents and associated casualties is largely compiled by the police, hospitals and other legal reporting agencies. In the UK, a standard STATS19 form is used by the police to report these data, after an incident has been reported by the public, and that report is used for the STATS19 dataset which is made publicly available on the data repository of the UK government. A limitation of this dataset is the unavailability of the records of all the accidents or casualties in Great Britain since every incident is not reported by the general public and thus, they are not reported by the police. In addition to that, accidents occurred on private land or car parks are not included in this dataset (Robineau, 2019). Despite the limitations of the dataset, it is still the most detailed, complete and reliable single source of information on road accidents and casualties in Great Britain as stated by Robineau (2019), and that was the most important reason for the selection of this dataset for this research project. As mentioned earlier, the data comprises of separate datasets for accident, casualty and vehicle each. The accident dataset contains features like road surface condition, light condition, weather condition and special condition at the site which were explored to answer one of the research objectives involving the identification of key factors during the majority of the accidents. The data for the years from 2015 to 2018 was selected for this project since older data might not be coherent with the current road accident situation.

### 3.6.2. Data Description

To get a better understanding of the dataset, it was important to know the meaning of each feature present in the data. For the road safety dataset used for this research project, a data guide was made available at the UK government’s data repository which was referred to comprehend the meaning of each column in the dataset. The description of the features used for the descriptive analytics and predictive analytics are presented in Table 3.6.2.1.

|  |  |
| --- | --- |
| Feature name | Feature description |
| Accident Index | Unique id associated with the accident |
| Location Easting OSGR | Eastward measured geographical cartesian coordinate of the accident site |
| Location Northing OSGR | Northward measured geographical cartesian coordinate of the accident site |
| Longitude | Longitude of the accident site |
| Latitude | Latitude of the accident site |
| Police Force | Numerical value denoting one of the 51 police forces associated with the accident |
| Accident Severity | Numerical value denoting the severity of the accident among fatal, serious or slight (1, 2 or 3) |
| Number of Vehicles | Count of vehicles associated with the accident |
| Number of Casualties | Count of casualties associated with the accident |
| Date | Date of the accident in the DD/MM/YYYY format |
| Day of Week | Numerical value denoting one of the 7 days of the week |
| Time | Time of the accident in the HH:MM format |
| Local Authority (District) | Numerical code denoting one of the 416 districts associated with the accident |
| Local Authority (Highway) | Alpha-numeric code denoting one of the 207 highways associated with the accident |
| 1st Road Class | Numerical value denoting one of the 7 classes of the 1st road |
| 1st Road Number | Road number of the 1st road |
| Road Type | Numerical value denoting one of the 8 types for the road where the accident occurred |
| Speed limit | The speed limit of the road associated with the accident |
| Junction Detail | Numerical code denoting the details about the junction associated with the accident (-1 to 9) |
| Junction Control | Numerical code denoting the control present at the junction associated with the accident (-1 to 4) |
| 2nd Road Class | Numerical value denoting one of the 7 classes of the 2nd road |
| 2nd Road Number | Road number of the 2nd road |
| Pedestrian Crossing-Human Control | Numerical value denoting one of the 4 types of human control at the pedestrian crossing |
| Pedestrian Crossing-Physical Facilities | Numerical value denoting one of the 7 types of physical control at the pedestrian crossing |
| Light Conditions | Numerical value denoting one of the 6 conditions of lighting associated with the accident |
| Weather Conditions | Numerical value denoting one of the 10 conditions of weather associated with the accident |
| Road Surface Conditions | Numerical value denoting one of the 8 conditions of road surface associated with the accident |
| Special Conditions at Site | Numerical value denoting one of the 9 special conditions at the site of the accident |
| Carriageway Hazards | Numerical value denoting one of the 9 types of carriageway hazards associated with the accident |
| Urban or Rural Area | Numerical code denoting the area type of the accident location among urban, rural or unallocated (1, 2 or 3) |
| Casualty Gender | Derived column from the casualty data to represent the gender of the majority of the casualties for an accident (Majority Male, Majority Female or Both Gender) |
| Average Casualty Age | Derived column from casualty data to represent average age of casualties for an accident |
| Hour | Derived column from the time column to denote the hour of the accident |
| Min | Derived column from the time column to denote the minute of the accident |

1. : Description of the required features present in the data

### 3.6.3. Data Exploration

The data was available yearly so, there were 4 CSV files for the four years for each category i.e., accident, casualty and vehicle. Each file was read as a DataFrame in Python and then the four accident DataFrames were merged to create a single DataFrame for accidents. Similarly, one DataFrame for casualties and vehicles each were also made. Each DataFrame contained a mixture of continuous, categorical and nominal columns. A big advantage of this dataset that was noticed immediately was the huge volume of the data related to accidents and casualties which meant that the machine learning models could be trained properly and small irregularities in the data like missing values could be ignored without losing too much information. The categorical and nominal columns were label-encoded to represent the data as numbers. To understand the meaning of the label-encoded data, a data guide was also made available at the UK government’s data repository. The descriptive statistics of each column in the dataset was inspected to summarise them and gain an overview of the values in each of them.

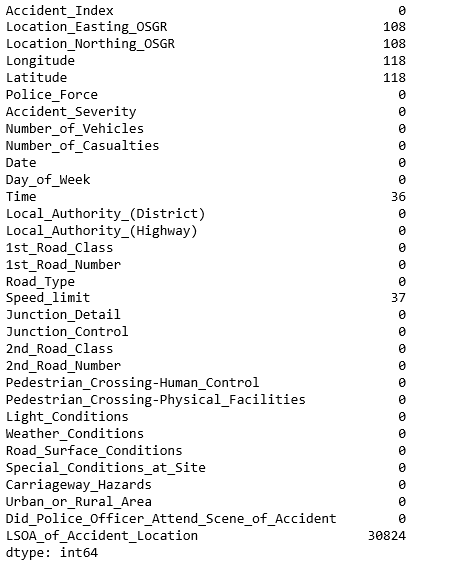
## 3.7. Data Preparation

The data preparation phase of the research methodology involved the construction of a version of the dataset that was appropriate for further analysis with the help of various data wrangling techniques. Many columns that were derived from existing features were added in the dataset to make those data suitable for the machine learning algorithms. The data preparation comprised of the following steps:

* Data Cleaning
* Data Merging
* Data Transformation
* Data Construction

### 3.7.1. Data Cleaning

There is a popular term in the field of computer science that says, ‘garbage in, garbage out’. It refers to the concept that the quality of the output of a computer program depends upon the quality of the input provided to it. In terms of data analytics, it can be interpreted as the quality or the credibility of the output of analysis depends upon the quality of the input variables used for the analysis. Erroneous input data may present a misleading picture of the actual scenario which could lead to the adoption of incorrect measures. One common flaw in many datasets is the missing values. Depending on the feature column in which data is missing and its relationship with other features, suitable imputation techniques could be employed to fill the missing values. In this dataset, the column ‘LSOA\_of\_Accident\_Location’ had data only for England and Wales and was not of much significance to this project, hence, this column was dropped from the DataFrame. The count of missing values in other columns, as depicted in Figure 3.7.1.1, was insignificant compared to the total volume of the data, so the rows with the missing values were simply removed from the DataFrame.



1. Count of missing data in the accident DataFrame

### 3.7.2. Data Merging

Data merging refers to the process of bringing together data from multiple sources. As mentioned earlier, the road safety data were available as separate files for each year, so these files were merged into a single file for ease of analysis. The information regarding the casualties of road accidents was available in a separate DataFrame. To make predictions about the gender of the casualties and the average age of the casualties, these data were merged from the casualty DataFrame to the accident DataFrame through the accident index which acted as a common link between the two DataFrames.

### 3.7.3. Data Transformation

To merge the information about the gender of the casualties and average age of the casualties for an accident, these data were transformed from their original representation in the casualty DataFrame into a summarised version of themselves. The count of male and female casualties for individual accidents was calculated and compared to transform them from numbers into three categories named ‘Majority Male’, ‘Majority Female’ and ‘Both Gender’. The gender of casualties for the accidents where the count of male casualties was greater than the count of female casualties was represented by ‘Majority Male’. Similarly, the gender of casualties for the accidents where the count of female casualties was greater than the count of male casualties was represented by ‘Majority Female’. The gender of casualties for the accidents with an equal number of male and female casualties was represented by ‘Both Gender’.

For the regression tasks of the predictive analytics, it was required to transform the categorical and nominal columns, which were label-encoded by default, into dummy-encoded columns or one-hot encoded columns. This transformation was necessary to avoid the bias towards values represented by bigger numbers in a label-encoding. However, a disadvantage of dummy-encoding or one-hot encoding is the significant increase in the number of columns in case of columns with a large number of categories. Dummy encoding was selected as the transformation strategy for this dataset since it is easy to implement, and it is available with the ‘Pandas’ library in Python which was already imported to read the CSV files into the DataFrames.

### 3.7.4. Data Construction

The time column represented the time of the accident in HH:MM format. The data in this format is not of much use to the algorithms. Therefore, the time data was split into two separate columns, one for the hour and the other for the minute. Another major issue with the dataset related to the prediction of accident severity was the class imbalance. More than 80% of the data belonged to ‘Slight’ accident severity class which was represented by number 3 in the dataset whereas, merely 16.5% of the data belonged to the ‘Serious’ accident severity class which was represented by number 2 in the dataset and only 1.3% of the data belonged to ‘Fatal’ accident severity class which was represented by number 1 in the dataset. If this data is used as it is for training a machine learning model for prediction of accident severity, then there is a high chance that the model would be biased towards the majority class. To resolve this issue of class imbalance, two techniques were tested in the experiment. One technique was known as Random Under Sampler which involved random removal of samples for the classes to bring the count of samples for each class close to the count of the minority class. Another technique was known as SMOTE (Synthetic Minority Oversampling Technique) which created synthetic samples for the classes to bring the count of samples for each class close to the count of the majority class. The issue of class imbalance also arose during the prediction for casualty gender where close to 60% of the data belonged to ‘Majority Male’ gender class, 35% data belonged to ‘Majority Female’ gender class and only 7.6% data belonged to ‘Both Gender’ class. The same approach was taken in that case as well concerning the imbalance among the classes.

## 3.8. Modelling

A machine learning model learns from the given set of input variables and creates a mathematical model to predict the value of the desired target variable. As discussed in chapter 2, if the target variable is a categorical feature, then the model aims to predict the class of the future data and this subset of machine learning is called classification. Whereas, if the target variable is a continuous feature, then the model aims to predict the exact value of the target for future data, and this subset of machine learning is called regression. However, the common requirement for both these techniques is the availability of a set of training data to train the model and a set of testing data with known outcomes to test the model and evaluate its performance. For the models used in this research project, 70% of the data was used to train the model and 30% of the data was used to test the model. A random state of 42 was used for the train-test split.

### 3.8.1. Modelling Technique Selection

Several machine learning algorithms can be used for one predictive task. The selection of a suitable modelling technique often depends on the type of data that is being used. In this project, the majority of the columns contained label-encoded data and therefore, tree-based algorithms such as Decision Tree and Random Forest were preferred wherever possible since they work well with categorical data. Naïve Bayes was also selected for the classification tasks since it works with probabilities and thus, it is suitable for data with a large number of categorical features.

For the regression tasks, Random Forest Regressor was selected as one of the models because of the above-mentioned reasons. Linear Regression was also selected for regression tasks. For proper implementation of linear regression, the categorical features were dummy encoded to avoid the bias towards those classes which were represented by large numbers. Another model that was selected for regression was K-Nearest Neighbour Regressor. This algorithm works with the distance between the data points, so dummy encoded data was used for this as well.

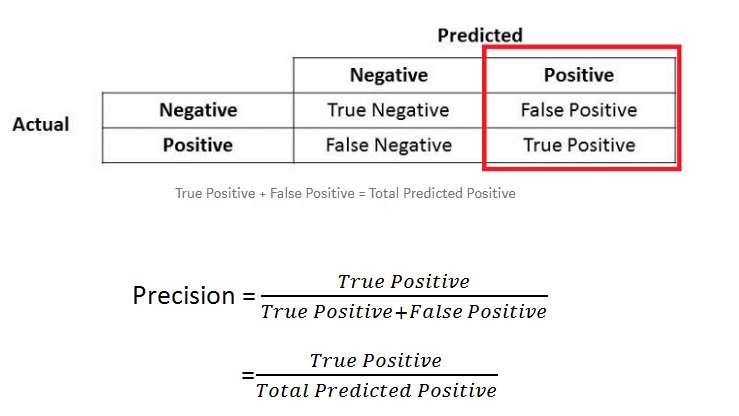
### 3.8.2. Model Building

The discussed earlier, the selected models for all the predictive objectives were built using the scikit-learn (sklearn) package in Python. The models were trained on 70% of the data and the remaining 30% were used for testing the model. Default parameter settings were set for the initial models. The optimal value of neighbours (k) for the KNN model was decided after running several iterations of the model with different k values. The k value for which the lowest error was observed was selected.

The columns which were not considered for predictive models were: Accident\_Index, Date, Did\_Police\_Officer\_Attend\_Scene\_of\_Accident and Local\_Authority\_(Highway). For the prediction of accident severity, additional columns that were dropped from the data were: 'Number\_of\_Vehicles', 'Number\_of\_Casualties', 'Casualty\_Gender' and 'Avg\_Casualty\_Age'. These columns were removed because this information would be unavailable in a test case since each of them is already a predictive objective of this project. The prediction of casualty count and vehicle count for an accident comes after the prediction of accident severity and therefore, accident severity was considered as an input variable for their prediction. For the prediction of casualty gender, it was assumed that accident severity and the number of casualties will be known. Thus, those input variables were considered during the prediction of casualty gender. However, the average age of casualty would be unknown at this stage so 'Avg\_Casualty\_Age' was not considered as an input. The prediction of 'Avg\_Casualty\_Age' was the last predictive objective. Since all other predictions would be made at this point, all the input variables were considered in this case.

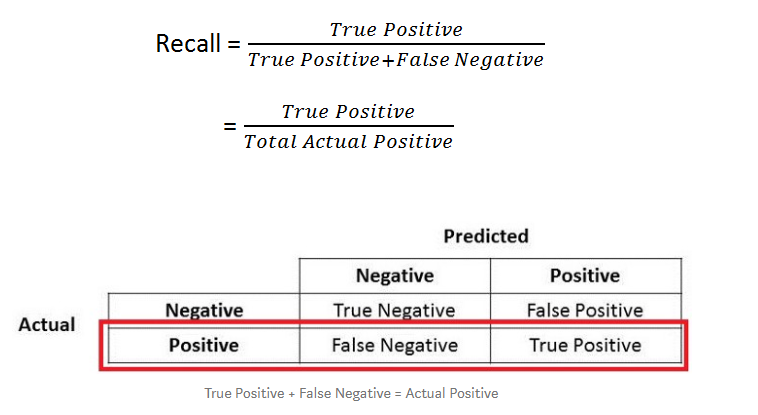
## 3.9. Model Evaluation

In this phase of the research methodology, the models were tested, and their performance was evaluated. The models were also judged in terms of their contribution towards the intended research objective. The classification tasks of the project were the prediction of accident severity and the gender of the casualties. The key evaluation metric that is generally used for classification tasks is the accuracy score. However, for imbalanced datasets, it is easy to obtain high accuracy by predicting every outcome to be of the majority class. Therefore, apart from the accuracy score, confusion matrix and classification report were also inspected to identify the best model for the task. Through the confusion matrix, the exact number of predictions for each class was viewed and compared against the actual number of test cases for each class. In the classification report, the model statistics such as the accuracy, precision, recall and f1-score was checked. The accuracy determines the percentage of time a model correctly predicts a class. It is a useful metric in case of a balanced dataset. For an imbalanced dataset, precision, recall and f1-score are more suitable metrics.



1. Precision formula (Shung, 2018)

Precision indicates the number of times the model was able to correctly predict a class out of all the predictions of that class.



1. Recall formula (Shung, 2018)

Recall indicates the number of times the model was able to correctly identify a class out of all the actual test samples of that class. It was the primary performance metric for the prediction of accident severity class since false negatives were more dangerous than false positives for that task.



1. F1-score formula (Shung, 2018)

F1-score is an overall indicator of the model’s performance which incorporates both precision and recall.

The regression tasks of the project were the prediction of the number of casualties, number of vehicles and the average age of the casualties. The performance of a regression model is usually measured based on the root mean square of the errors and the R-squared (R2) score. The RMSE is an absolute measure of fit since it indicates the exact errors in the predictions compared to the actual values, whereas the R2 score is a relative measure of fit that indicates the proportional improvement in the model compared to a model that makes all the predicted values equal to the mean of the target variable (Martin, n.d.). The R2 score also represents the proportion of the target variable that can be explained by the independent variables. For the linear regression model, the shape of the distribution of the residuals was also examined to check if it follows a normal distribution or not, along with the homoscedasticity of the residuals. The normal distribution and homoscedasticity of the residuals indicates a good fit of the model.

## 3.10. Limitations of the methodology

The main limitation of the methodology adopted for this research project was the lack of focus on the vehicle’s data related to the accidents as it was relinquished in exchange for an in-depth analysis of the accident and the casualty’s data. Another limitation of the research methodology was the lack of analysis regarding the trend of accidents over a period of time. Missing or unknown data were not considered for the analysis due to the abundance of known data. However, it could have revealed some interesting information.

From the prediction’s point of view, the most important limitation of the project was the transformation of certain target variables to fit them as per the algorithm’s requirements. For example, instead of predicting the individual gender of each casualty associated with an accident, the model predicted that gender which belonged to the majority of the casualties for that accident. Similarly, instead of predicting the age of all the casualties associated with an accident, the model predicted the average age of the casualties for that accident. The nature of the data also put limitations on the choice of machine learning algorithms. For example, distance-based algorithms for classification were rejected in favour of tree-based and probability-based algorithms due to the presence of high-order label encoded columns in the data.

# Chapter 4 – Findings and Analysis

## 4.1. Introduction

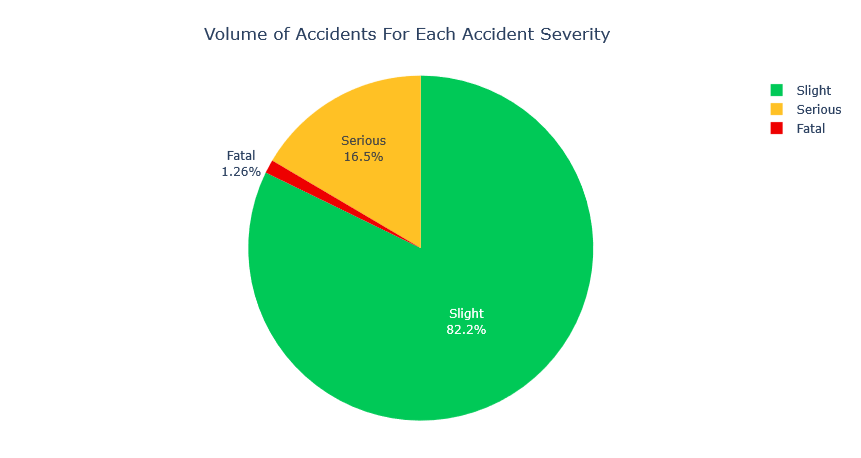
This chapter presents the results of the analysis and the outcomes of all the predictive models. The findings of the descriptive and exploratory analysis are discussed with the help of various visualizations like pie charts and bar graphs. The summary of the predictive models along with their evaluation metrics is also laid out in this chapter.

## 4.2. Data Visualization and Analysis

Data visualization is a powerful tool for easier and better understanding of the underlying message in the data. Visualization of important data was carried out in this project to identify the factors that had a major impact on fatal accidents or accidents in general. The infrastructural factors in the data associated with the accidents were the light conditions, the road surface conditions and the special conditions at the site. The environmental factors in the data associated with the accidents were the weather conditions. The overall count of accidents and the count of fatal accidents were studied and visualized for each sub-category belonging to these categories to recognize the most impactful condition from each category. It was crucial to not only look at the count of total accidents and fatal accidents for each category but also to evaluate the percentage of accidents that were fatal in each category. This was needed to remove the misinterpretation of certain sub-categories as key reasons for fatal accidents just because they were present during most of the accidents.

### 4.2.1. Accident Severity

Since the prediction of accident severity was one of the objectives of this project, therefore, the distribution of data for each severity class was explored with the help of a pie chart which can be seen in Figure 4.2.1.1.

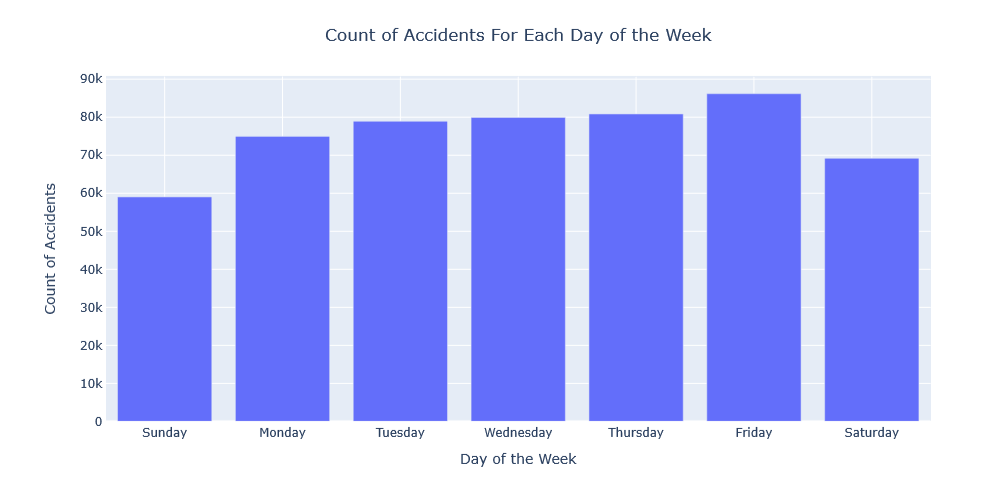


1. Distribution of accidents for each severity class

It was observed that most of the data samples i.e., 82.2% of the samples belonged to the ‘Slight’ severity class and only 1.26% of the data belonged to ‘Fatal’ severity class. This imbalance in the target variable causes machine learning algorithms to get biased towards the class with a larger share of the samples and therefore, should be handled properly before training the models. The steps taken by the author to deal with this problem was discussed in the ‘Data Preparation’ section of chapter 3.

### 4.2.2. Day of the Week

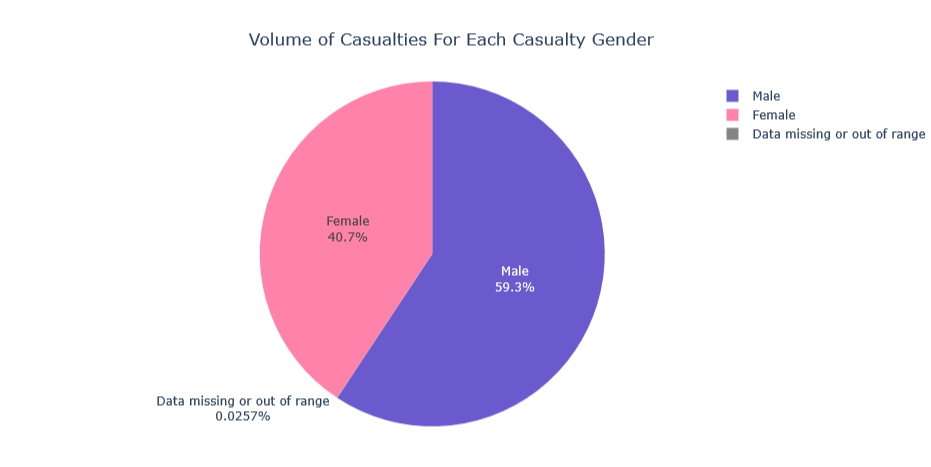
The count of accidents on each day of the week was visualized to check if the number of accidents was higher or any specific day (see Figure 4.2.2.1). It was observed that the maximum number of accidents occurred on a Friday and the minimum number of accidents occurred on a Sunday, however, the numbers for each day were not significantly different from each other. This indicated that day of the week did not influence the number of accidents.



1. Count of accidents for each day of the week

### 4.2.3. Casualty Gender

The distribution of casualty’s gender was visualized with the help of a pie chart to understand the distribution of samples in each gender class. The visualization depicted in Figure 4.2.3.1 shows that majority of the casualties were male. The proportion of unavailable gender data was negligible, so it was not considered for further analysis.

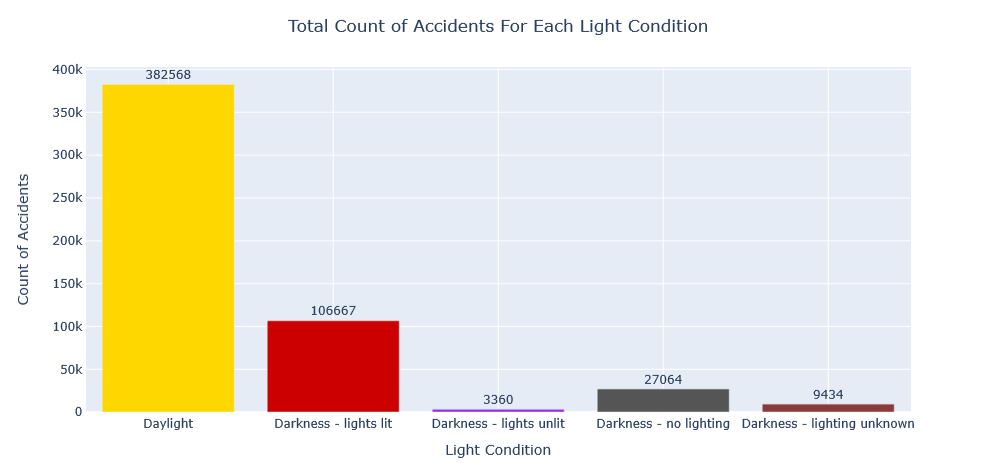


1. Distribution of casualties for each gender class

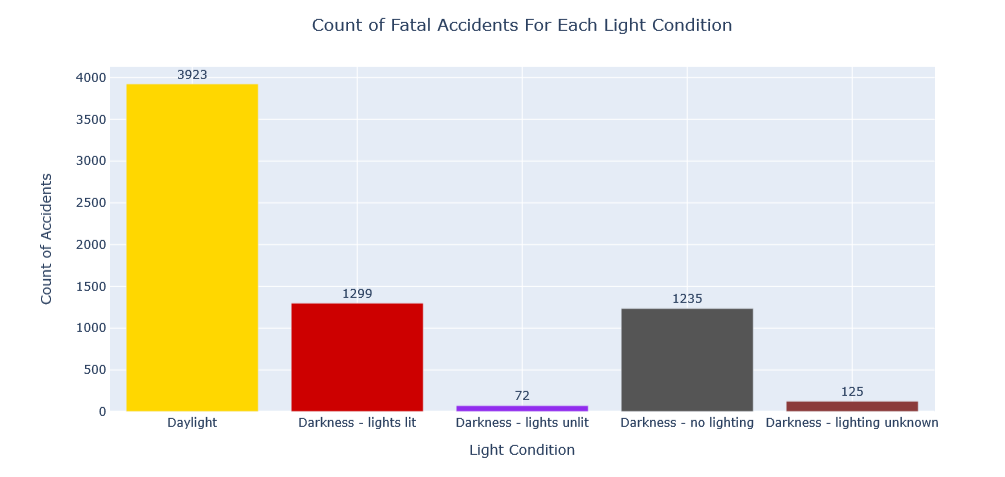
### 4.2.4. Light Conditions

The light condition column in the dataset represented the condition of light during the accident. Its analysis allowed the understanding of key light conditions during the majority of the fatal accidents and accidents in general. The sub-categories for the light condition were:

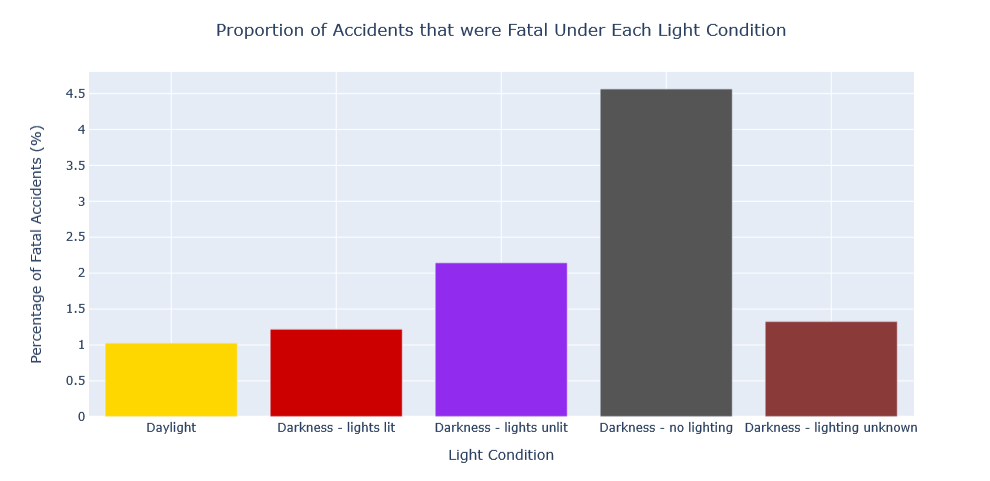
* **Daylight:** Denotes the light condition for the accidents that occurred during daytime
* **Darkness – lights lit:** Denotes the light condition for the accidents that occurred during night-time with the presence of artificial light in the vicinity
* **Darkness – lights unlit:** Denotes the light condition for the accidents that occurred during night-time when the lights in the vicinity were unlit
* **Darkness – no lighting:** Denotes the light condition for the accidents that occurred in the darkness with no facility for lighting
* **Darkness – lighting unknown:** Denotes the light condition for the accidents that occurred during darkness with unknown lighting condition



1. Total count of accidents for each light condition



1. Count of fatal accidents for each light condition



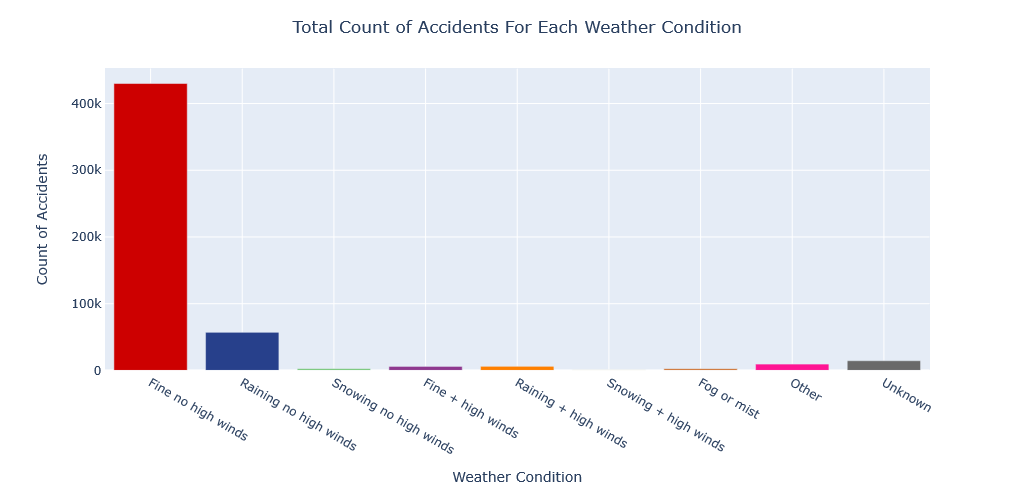
1. Proportion of accidents that were fatal under each light condition

From the above figures, it can be seen that the maximum number of accidents happened during daylight and therefore the maximum number of fatal accidents also occurred during daylight, but it was because of the volume of the traffic during daylight. This may give a false impression that daylight is the worst light condition for road journeys. However, it was only when the proportion of fatal accidents among the total accidents for each light condition was checked, it was observed that among all the light conditions, the highest percentage of fatal accidents occurred during darkness with no lighting followed by darkness with lights unlit. This information could be used to reduce the areas with these light conditions because there is a higher chance of an accident being fatal in areas with such light conditions.

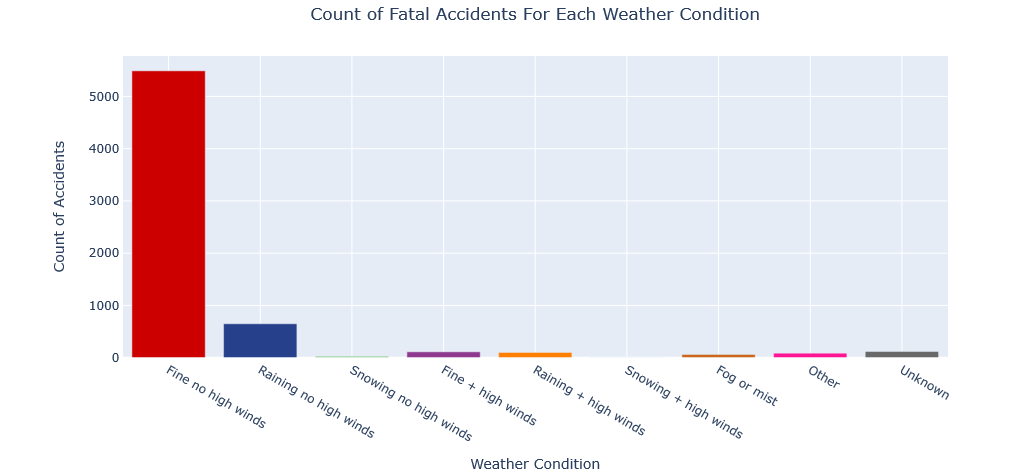
### 4.2.5. Weather Conditions

The weather condition column in the dataset represented the condition of weather during the accident. Its analysis allowed the understanding of key weather conditions during the majority of the fatal accidents and accidents in general. The sub-categories for the weather condition were:

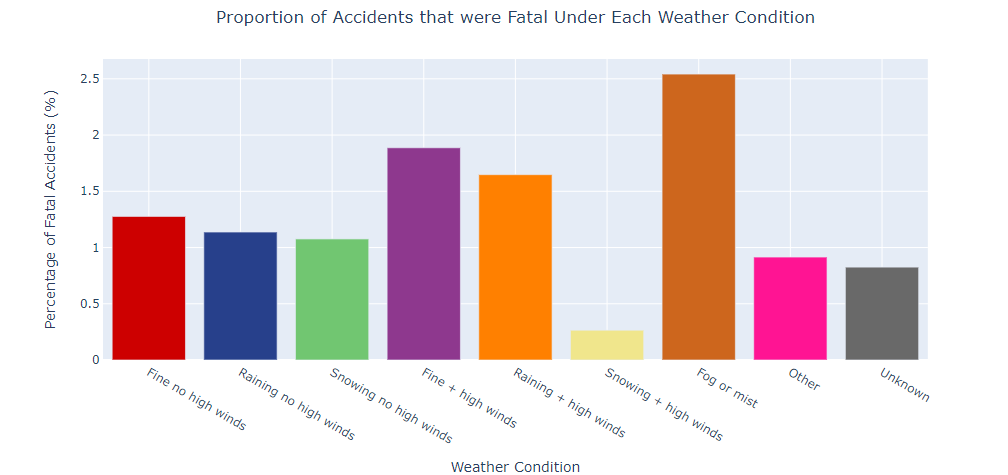
* **Fine no high winds:** Denotes a clear weather condition without any strong wind
* **Raining no high winds:** Denotes a rainy weather condition without any strong wind
* **Snowing no high winds:** Denotes a snowy weather condition without any strong wind
* **Fine + high winds:** Denotes a clear weather condition with strong winds
* **Raining + high winds:** Denotes a rainy weather condition with strong winds
* **Snowing + high winds:** Denotes a snowy weather condition with strong winds
* **Fog or mist:** Denotes a weather condition with obstructed visibility due to fog or mist
* **Other:** Denotes a different weather condition that those listed above
* **Unknown:** Denotes an unknown weather condition during the accident



1. Total count of accidents for each weather condition



1. Count of fatal accidents for each weather condition



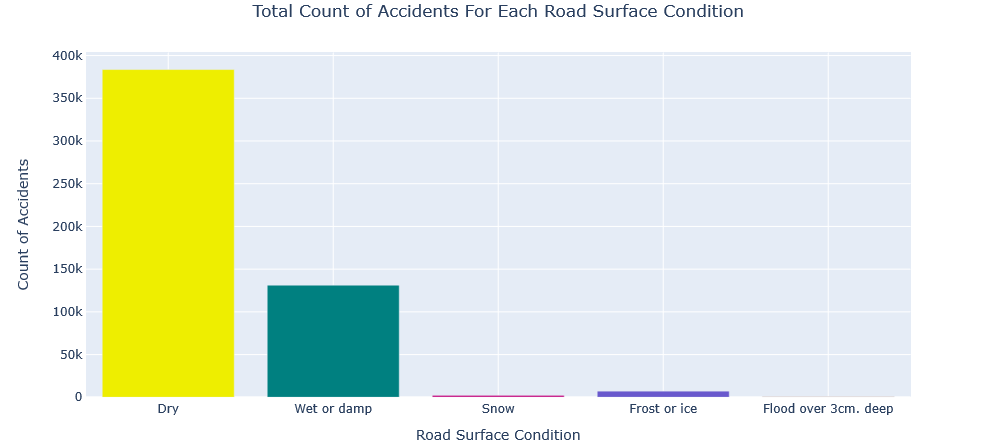
1. Proportion of accidents that were fatal under each weather condition

From the above figures, it can be seen that the maximum number of accidents happened during fine weather with no high winds and therefore the maximum number of fatal accidents also occurred during fine weather with no high winds, but it was because of the volume of the traffic during that weather. This may give a false impression that fine weather is the worst weather condition for road journeys. But like the light condition, when the proportion of fatal accidents among the total accidents for each weather condition was checked, it was observed that the highest percentage of fatal accidents occurred during foggy or misty weather condition. This information could be used to spread awareness among the commuters to be cautious when the weather is foggy or misty, and to encourage the vehicle manufacturers to work on technologies to reduce the negative impact of fog or mist on visibility.

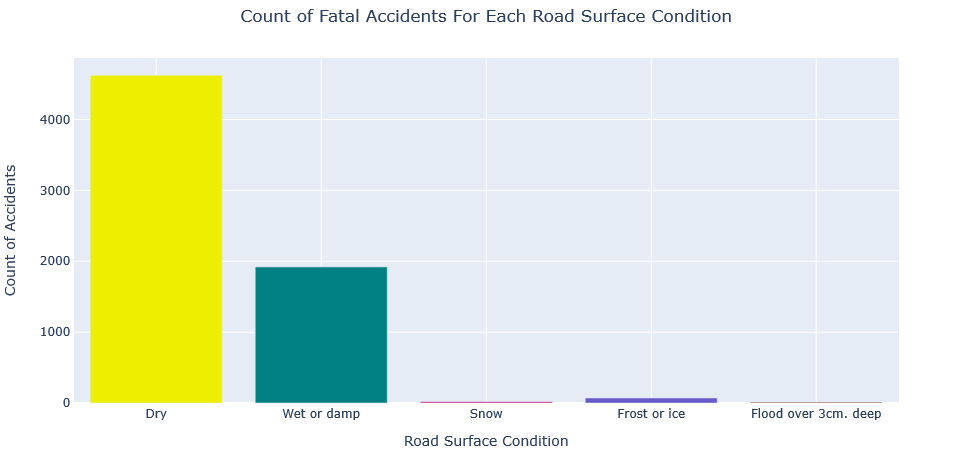
### 4.2.6. Road Surface Conditions

The road surface condition column in the dataset represented the condition of road surface during the accident. Its analysis allowed the understanding of key road surface conditions during the majority of the fatal accidents and accidents in general. The sub-categories for the road surface condition were:

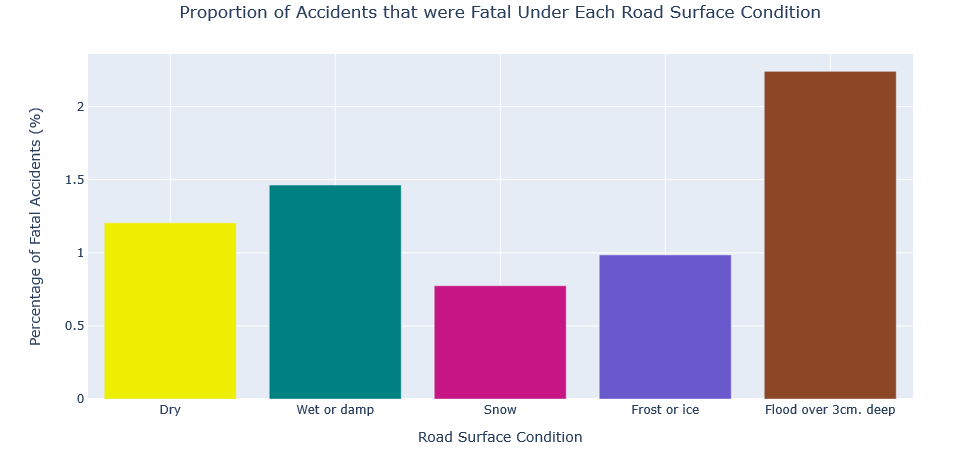
* **Dry:** Denotes a dry road surface with negligible traces of degradations like wetness or snow
* **Wet or damp:** Denotes a road surface with the presence of water or dampness up to 3 cm
* **Snow:** Denotes a snowy road surface condition
* **Frost or ice:** Denotes a road surface with the presence of ice
* **Flood over 3cm deep:** Denotes a road surface with flood water deeper than 3 cm



1. Total count of accidents for each road surface condition



1. Count of fatal accidents for each road surface condition



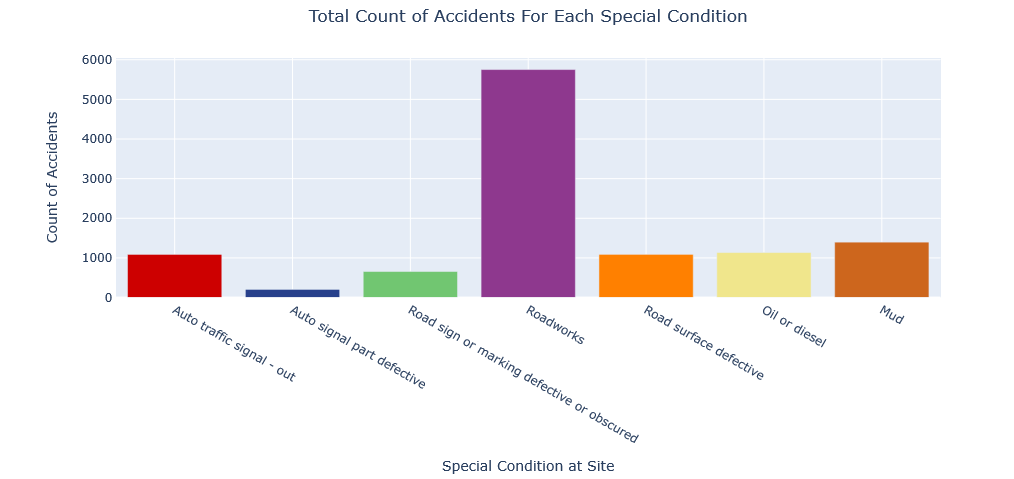
1. Proportion of accidents that were fatal under each road surface condition

From the above figures, it can be seen that the maximum number of accidents happened during dry road surface condition and therefore the maximum number of fatal accidents also occurred during dry road surface condition, but it was because of the volume of the traffic during that road surface condition. This may give a false impression that dry road is the worst road surface condition in terms of accidents. But like previous conditions, when the proportion of fatal accidents among the total accidents for each road surface condition was checked, it was observed that the highest percentage of fatal accidents occurred when road surface was flooded with more than 3 cm deep water. This information could be used as a motivation for the creation of new technologies for prevention of water clogging on the road surface.

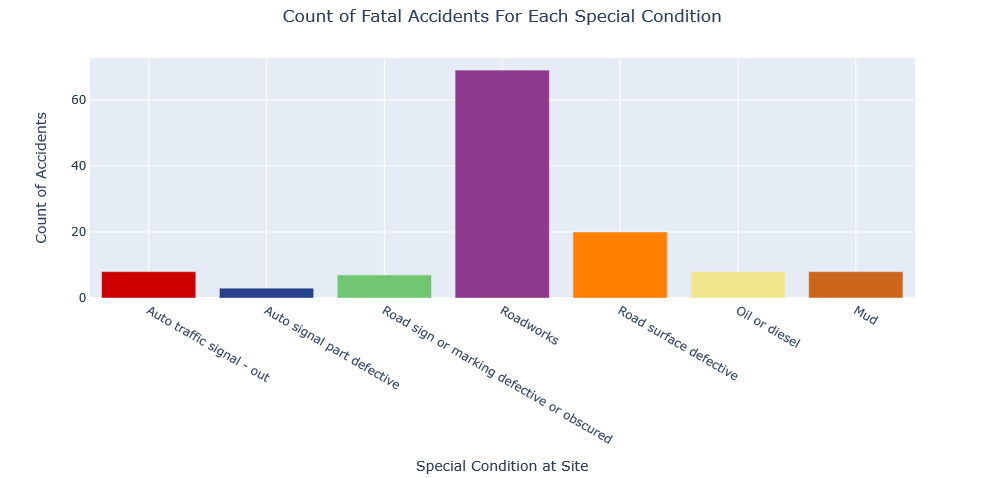
### 4.2.7. Special Conditions at Site

The special condition at site column in the dataset represented an unusual condition at the accident site which did not get covered by previous columns. Its analysis allowed the understanding of key special conditions during the majority of the fatal accidents and accidents in general. The sub-categories for the special condition at the site were:

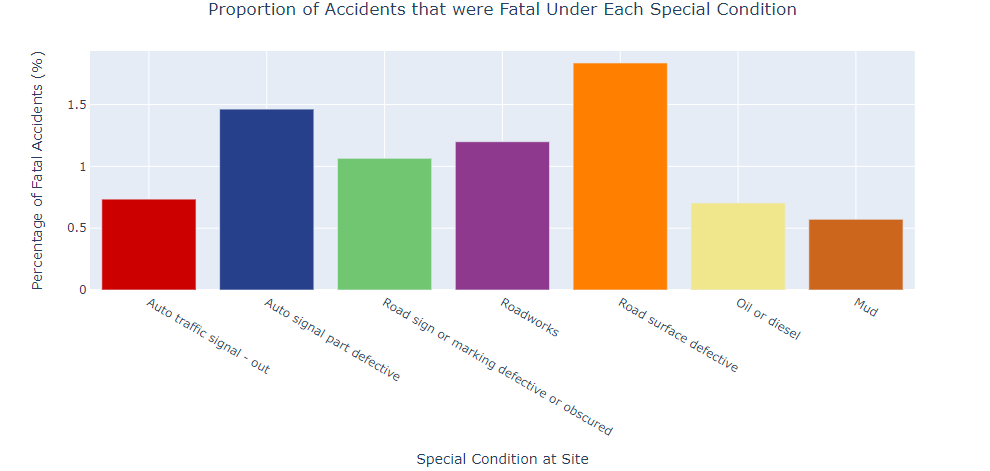
* **Auto traffic signal - out:** Denotes a complete outage in the automatic traffic signal at the accident location
* **Auto signal part defective:** Denotes a partially defective automatic traffic signal at the accident location
* **Road sign or marking defective or obscured:** Denotes a defective or illegible road sign or marking at the accident location
* **Roadworks:** Denote ongoing roadwork at the accident location
* **Road surface defective:** Denotes a defective road surface at the accident location
* **Oil or diesel:** Denotes the presence of oil or diesel at the accident location
* **Mud:** Denotes the presence of mud at the accident location



1. Total count of accidents for each special condition



1. Count of fatal accidents for each special condition

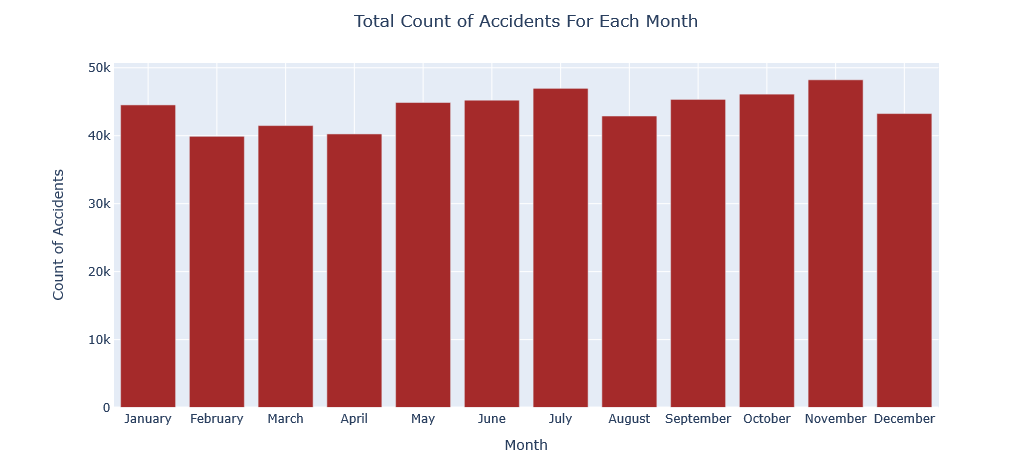


1. Proportion of accidents that were fatal under each special condition

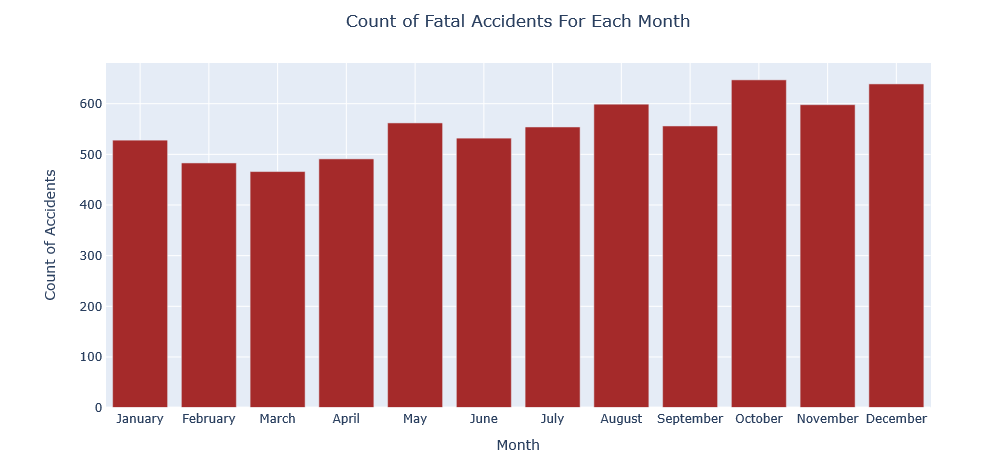
From the above figures, it can be seen that the maximum number of accidents happened during the presence of roadwork and therefore the maximum number of fatal accidents also occurred during the presence of roadwork, but it was because of the volume of the traffic. This may give a false impression that roadwork is the most dangerous special condition for an accident. But like previous conditions, when the proportion of fatal accidents among the total accidents for each special condition was checked, it was observed that the highest percentage of fatal accidents occurred when the road surface was defective, followed by when the automatic traffic signal was partially defective. This information could be used to ensure that road surfaces are periodically assessed and repaired in case of any damage along with the deployment of more reliable traffic signals.

### 4.2.8. Month

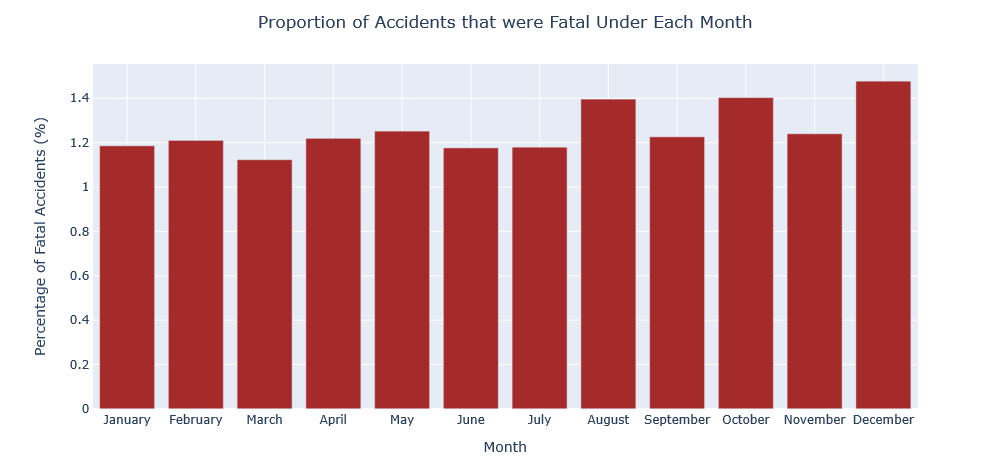
The count of total accidents and fatal accidents for each month was plotted for identification of those months when most of the accidents happen.



1. Total count of accidents for each month



1. Count of fatal accidents for each month

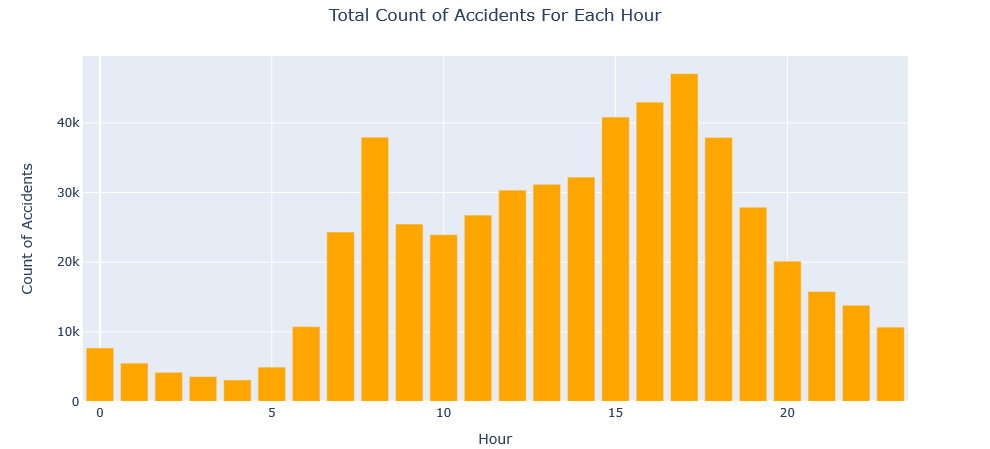


1. Proportion of accidents that were fatal under each month

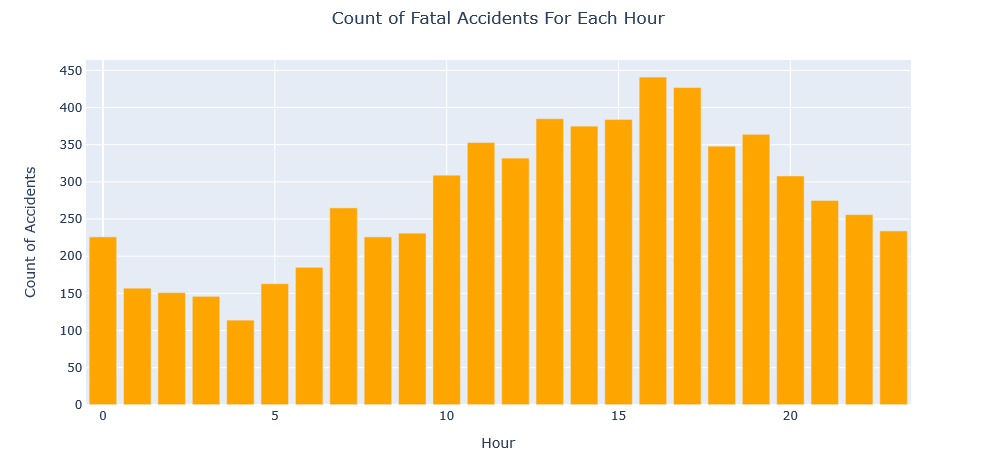
From the above figures, it can be seen that number of accidents was almost evenly distributed throughout the year. However, fatal accidents seemed to be happening slightly more during the second half of the year. In terms of the proportion of accidents being fatal, August, October and December had higher chances of an accident turning out to be fatal. Thus, it can be inferred that these three months are a bit deadlier than the rest of the year as far as road casualties are concerned.

### 4.2.9. Hour

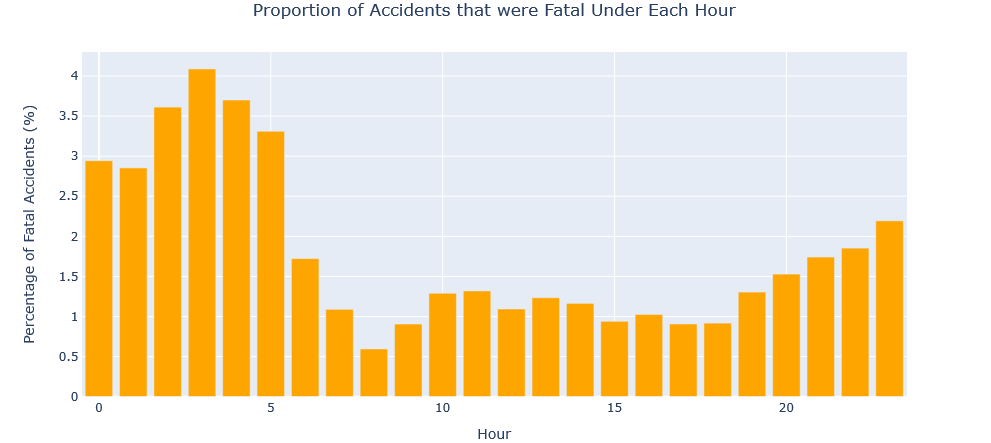
The count of total accidents and fatal accidents for each hour was plotted for identification of those hours when most of the accidents happen.



1. Total count of accidents for each hour



1. Count of fatal accidents for each hour



1. Proportion of accidents that were fatal under each hour

From the above figures, it can be seen that most of the accidents happened at 8 am and between 3 to 5 pm which is quite apparent since the volume of traffic is very high at these hours due to school and work timings. The hours for fatal accidents are more spread out and are on the higher side between 11 am to 7 pm. The most dangerous hours are between midnight to 5 am as the chances of an accident resulting in a fatality are extreme. This could be because of drowsiness among drivers during these hours. Thus, commuters should be extremely cautious when driving late at night or very early morning.

### 4.2.10. Top 5 Districts and Highways

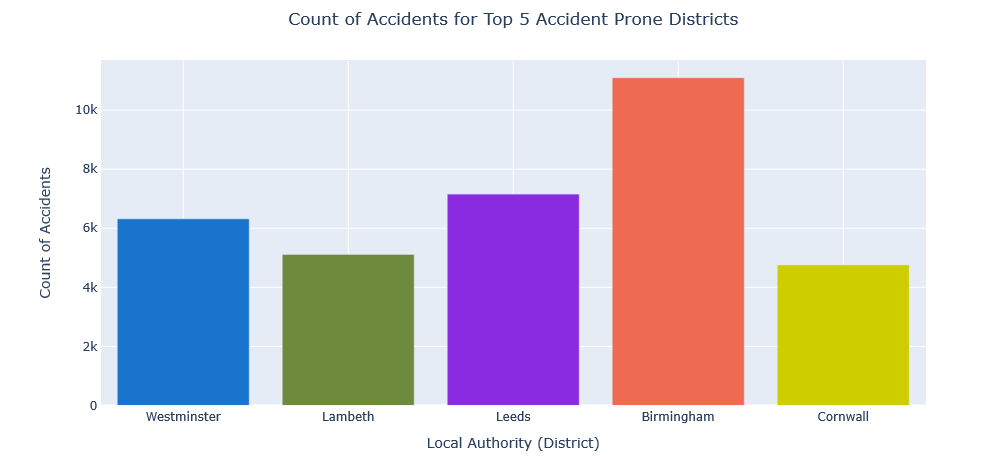
The dataset contained accident information for 416 districts and 207 highways of Great Britain identified by the local authority in whose area the accident occurred. The volume of accidents for every accident location in the dataset can be inferred with the help of Figure 4.2.10.1.

![A close up of a map

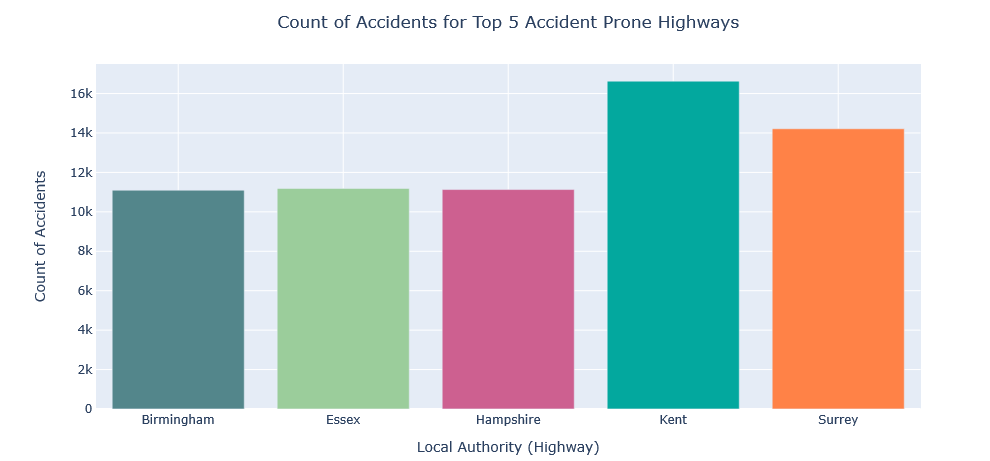
Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RD4RXhpZgAATU0AKgAAAAgABAE7AAIAAAAPAAAISodpAAQAAAABAAAIWpydAAEAAAAeAAAQ0uocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEFkaXR5YSBDaGFuZHJhAAAABZADAAIAAAAUAAAQqJAEAAIAAAAUAAAQvJKRAAIAAAADNDkAAJKSAAIAAAADNDkAAOocAAcAAAgMAAAInAAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADIwMjA6MDY6MTcgMTY6NTc6MjAAMjAyMDowNjoxNyAxNjo1NzoyMAAAAEEAZABpAHQAeQBhACAAQwBoAGEAbgBkAHIAYQAAAP/hCyFodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIwLTA2LTE3VDE2OjU3OjIwLjQ4NjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5BZGl0eWEgQ2hhbmRyYTwvcmRmOmxpPjwvcmRmOlNlcT4NCgkJCTwvZGM6Y3JlYXRvcj48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAcFBQYFBAcGBQYIBwcIChELCgkJChUPEAwRGBUaGRgVGBcbHichGx0lHRcYIi4iJSgpKywrGiAvMy8qMicqKyr/2wBDAQcICAoJChQLCxQqHBgcKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKir/wAARCAILA90DASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD32GGW6nus3c0SxyhFVAmMbFPdSepNT/2dJ/0ELn/vmP8A+Io07/XX3/XwP/RaVdoApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+Io/s6T/AKCFz/3zH/8AEVdooApf2dJ/0ELn/vmP/wCIo/s6T/oIXP8A3zH/APEVdooApf2dJ/0ELn/vmP8A+IqGzZ2gbzHMjLLIm4gAkK5A6cdBWnWZZf6mT/r4m/8ARjUAPgieS3jdriTLKCcBfT6VJ9mb/n4l/Jf8KW1/48of+ua/yrOsfEC3xjddOvo7aZDJBdOiGOZcZBG1iy5HI3hfz4oA0Pszf8/Ev5L/AIUfZm/5+JfyX/Cqy6xbtYWF2Ek8u/MYiGBkbxkZ5/lmq1r4nsLvw9c6wnmrb2plEqMo3gxkgjAPU4yPUEetD0vfoG5pfZm/5+JfyX/Cj7M3/PxL+S/4VTi121nhtZIkmY3Nt9rCbRmOPbnLc4HUDGTk9OASI7LxHZahpNjqFus3lXsixIjKA8bnPDjPBGCD1/GnbW39dgvpc0Pszf8APxL+S/4UfZm/5+JfyX/CpqKQEP2Zv+fiX8l/wo+zN/z8S/kv+FNgvY7i7urdFYPauqOSOCSoYY/A1YoAh+zN/wA/Ev5L/hR9mb/n4l/Jf8Kmqvf3sWnadcXs4YxW8bSMEGSQBnj3oAd9mb/n4l/Jf8KPszf8/Ev5L/hVO91hrG0tJG067lnu5BElrGYvMVtrMQSXCcBT0Y/jUum6muorMptp7Se3cJLBPt3ISAw5VmU5BB4JoAn+zN/z8S/kv+FH2Zv+fiX8l/wqaigCH7M3/PxL+S/4UfZm/wCfiX8l/wAKmooAh+zN/wA/Ev5L/hTISTGdzFiGYZPfDEVZqtD9xv8Aro//AKEaAHad/rr7/r4H/otKu1S07/XX3/XwP/RaVYurmKzs5rm4bZFDG0jseygZJpN2VwWuhnxeILWXxJLoqxzCaNC3nFR5bMApKA5zuCupxjGD14NatebDXLa38P2epSWuqpqMN5/aM+/SLpVUOSJVLmPbhY2IBJx8grqdT8zVvEUGki7ntrMWhupTbSmN58sFVQ6kMqjknaQeV5xnL10XX/gX/r0DTfp/S/r1OgpGJCkgFiB0Heub1K2l0mzstM06/vEGo3wh+0TzmeSBNjOwV3yckRkAsTgt9BWhaaLDpKzy2t3qDhoiClzeyzjP94eYzFT9CB7Um7JsfVI0YJGmt45JIXgd1DNFIVLIfQ7SRkexI96q3OpfZ9YsbDyt32tZW37sbNgB6Y5zmuegvb258LeFrOO9lhm1SKNZ7vO6QKIDI2Cc/M23GTnGSetSf2Nb6V400Zrae+k8yK5DC6vprgdE5HmM2PwxVSVm0Stjq6K86t9Ug1a1mv7298TQahJJJ5H2OxvTBbAMQihUjMcmABksGyc9sAa00t/rUnhlZ7m+0p7u2llu4YCYmLbEJQhhleSfRh2Ipbq4+tjr6gs7yDULVbm0k8yFyQrbSM4JB4PuDXP2mnLbeIrzRlu9QksZ7FJist/M8iPvZSUlLb1yAOjduMc1D4H0G0ttFtL2ObUDKGlAWTUbh4/vsP8AVs5T9OvPWjfX+uwdbHQ6jqcOm/ZfPWRvtVylsmwA4Zs4JyenFO1LUItLsjc3CuyeZHHhACcu4QdSO7CsPxstw9to62UsUM51WDy3ljMiqfm5KhlJ/MVT8SW3iNNHVr/VdLmtxd22+OHTJInb9+nRjOwH5GiOv32/L/MHo/l+rOypk0qQQSTSnbHGpZjjOABk1gXcL634ouNOnu7q3srK2ilMVrcNA0zyFxkuhDYATgAjJJznAqhPasYNf0G6vL24tba0S5gkN1IsqBxIPLMikMwBjz8xJIbBzUybUWyoq8rHW288d1bRXEDbopUDo2CMgjIPNSVzXh7REsfDdvPpk959rmsFCG6v5541YoCDsd2Uc46DpxWfoxjsNQsl1hfEFjqEhEbPd3jz21zIRyOHaNcnoMIegAHStJK0nEzTvFSO1orlbbS/7a1/XxfajqQht7xI4Ybe+kgWP/R4mJHlkHkseCSO+Mk5rrqGovpcOlm+kE0mrPpxvwF8zy1Vn3dMbyq7M4684qd/69P8ynpr/Xf9Dsqggvbe6nuYYJN0lrIIphtI2sVVwOevysp49a5+Wybw5q+lmxvb6W3vbg209vd3clwDmN2DqZCzKQU6AgYJ44FU9B8M2P8Ab+vN5+qZg1CPZ/xNrrB/0eFvmHmfNyf4s8YHQAULV/15f5gdnRXF6rBcR6zez61Br0loXBtbnSruQRwx7Rw0UThmbcGJJRxgjntXW2U0Vxp9vNbT/aIZIlZJs58xSOG49etC1Vw6joJXmVzJbyQFXZQJCp3AHAYbSeD1GcH1AqWuNury8PgXXJY7yeOdL24SOZXO+MCcgYJ6YHAHSrN/pjaLdaZeW+palNPLexwTm4vHdJVfIOYs+WvqNqrgj8KF087fiD0v5X/A6d3SKNpJGCIoLMzHAAHc1hr4pMyCey0LVrqyIyt3FHGFZf7wRnEjD0whz2zTvGf/ACJ9/u/1W1fP/wCuW8eZ/wCObqs6wNUW0Eui3llbCNGZxc2jThwBxt2yJt7+tJuyuxpXdjRjcSxLIoYBlDAMpU8+oPI+hp1chf32ranaeFJLC6WyuNR+ad1BKqGt2Y4XoSDyobIyATnFJrzDQLTTdNivNW8q+umFzcx+dd3BVYyxCbQzLuKgfKAFG7GDzVSVm0TF3VzsKK5DQr6ODxCttpra3Jpstu7y/wBqW12BBIpXBEs65wwLZXcfujAHNY+q3ckGi3et6La+KJ2ihe5h1B79RAwALZMEkwGwgdos46c4pD30PQJbyCG8gtZJMTXAYxLtPzBcbueg6jrU9cjrmi2uo+MtGlnlv0aaGfcINQnhAwqYwEcBevOMZ75rqLS1jsrSO3haZkjGAZpnlc/V3JY/iaBXvqjEi8VzXUlx9g8N6tdwwXEluZ43tQrNGxVsBpg2Mg9q1tM1K31axW6td4QsyMkiFWRlJVlYHoQQRXJeHbbxDLZ6kdM1TT7e3Oq3u2OWwd5B+/fP7zzQP/HDj0NE9xc2vgaVNMLQalDqkMMzSTFg8zXUe8s6hcq27nAHBxgdKF0+X42/zKe/za/M7WeRoreSSOJ5mRSwijIDOQOg3EDJ9yB71FLfQW72iXJMUl3J5USEZJfYz7eMgcK3txWHf6dNo/hXWbptSvbm/ayld7h52AVgjEGOMHbHjttAPAySeapeItJt9SvvC8tzNeh5LoRt5N9NCMfZpmzhHADZ/iHOOM44o/4Aun3/AIHY0Vz2rW9lpmkW+nibV5POmxFDbXkslxOcElfNdtwXHOSygY6jpVDw/Pd2visaebLWLO0ns3mEWq3iXJ3o6DKOJZGAw/IJxwMd6Fq7f1tcHornVwSvMrmS3kgKuygSFTuAOAw2k8HqM4PqBUtcbdXl4fAuuSx3k8c6Xtwkcyud8YE5AwT0wOAOlWb/AExtFutMvLfUtSmnlvY4JzcXjukqvkHMWfLX1G1VwR+FC6edvxB6X8r/AIHU0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVmWX+pk/6+Jv/RjVp1mWX+pk/wCvib/0Y1AFi1z9ihx18tf5VzdjY3q6pCtvpdxpceW+2YuUa1lBVv8AVoGJBLkNnYhIBz6V0lp/x5w/9c1/lUtK1wOYsrXU3tdI0y402SBdNeMy3bSRmKURqVGwBi/Jx95VwM/Q07Pw7qUJgtWgC2dxie7HmL8ssZOwYzzu/d9OMRnPWuzopvXcNjlNKsp9NYWt5HslutJggiJII8yJGDR56Z+bI55+Y9jTRoGoW0miPaRL5e6A6hCXA2PHHgSDsTxtIHX5T2562inf3ub0f3aitpb+tTin0zVp9NgsP7LljNrHdIZnli2yl43VSuHJwSw+8BjP5SS+DLRrq426NaGGTTAoBjQhrkFvmI7vgj5z69a7Gipt/X3/AOZV/wCvu/yOHutBuppbzzdEM97NHELXUS8R+yuIlUvktvXDDPygk4qzb6DdL4pe7urS4kmFy8iX6tbqhiOdqE487gHbt+7xnNdfRTeruTbSxz+j6C2m+C0sbSKOx1GSyWOSVcFhN5eNzMM7iD3yenFY1z4Z+3abfwW3h2PT45LF4nt5PJIup8qUf5WIyuG+ZsN81dzRTvd3H0MXWNFS8j0m1t4mitLa5DMttKYDFGIpFG0oVIGSowv8s1i3fh28XSxZm1mvIIr5nmAeKSW9iKEIzedlWZcqp3kfcyOwrtKKQf1+f+Zx39hNFDaC80OTVLZIWSO0lkhZrVy5YNyVQfKQPkzt24GRzTrrRLmTUpGOleZePcRSW+qCRD9ljATKbmIk/hfgKQ27n7zY6+ijZ3Aw9E0pdJhuLhtPzeXF5MXdNpcxvOzKSxP3QrBtueOcDPFa0dxI8iq1pNGCXBZimF2nAJwxPzdR7dcHipqKACq0P3G/66P/AOhGrNVofuN/10f/ANCNADtO/wBdff8AXwP/AEWlP1LT4tV0+SzuWdYZSvmBCBuUEEqcjocYPsTTNO/119/18D/0WlWROhAI3kHodh/woAdJGksTRSKGR1KspHBB7VkSeGbdrGwhhu7y3uNPj8q3vYnXzlXABBypVgcDIZSMgHGRmtbzl9H/AO/bf4Uecvo//ftv8KAMw+H4p9Key1G9vL8vIJftE0gWRHGCrKYwoUggEbQP1NPsNImtGka61jUNQLpsAuTEAg9hGign3OTWh5y+j/8Aftv8KPOX0f8A79t/hQBmSeHLOTQbPSvMnVLFYxbzo+2WNkGFcMBjOOvGDkgjBxUdp4bEGqW+o3eq6hqF1bo6I1y6BdrYyNiIq9uuM+pNa/nL6P8A9+2/wo85fR/+/bf4UdbgY0nhnbNKdP1nUtOgmdpJLa2aIxlmOWI3xsy5JJ+Ujk561eGkwC6sJ98xaxjeOLfIXLBgASxbJY/KOc/XNW/OX0f/AL9t/hR5y+j/APftv8KPICAafENYbUtz+c0AgK5G3aGLZ6Zzk+tUtP0D+zL0yWmp3y2hd3FgxjMKsxJOCU3gZJON2B6Y4rU85fR/+/bf4Uecvo//AH7b/CjYCtqOmQ6l9l89pF+y3KXKbCBllzgHI6c07UtPi1SyNtcM6p5kcmUIByjhx1B7qKn85fR/+/bf4Uecvo//AH7b/CgDO1PQ01C7jvILy6069jQxi5tGXcyZztZXVlYZ5GVOOcYyafp2iW2nW9whea7lujuuZ7ltzzHGOcAADHAAAA7Cr3nL6P8A9+2/wo85fR/+/bf4UdLB5mVpvh86bA9sNW1G4s/K8mK3ldAIV6Da6oJMgcAliaZF4axcwyX2salqEVvIJYbe5aLYjj7pJRFZsdtxPPPXmtjzl9H/AO/bf4Uecvo//ftv8KOtw6WOXttDvrnxBr066rqemwy3iFUt0iCTKLeIFgXjY9QRlSOnqK128Oac2hrpSxyJbo29HWVhIsm7d5gfO7fu53ZyTnPWtHzl9H/79t/hR5y+j/8Aftv8KOlgMqy8PLb6hHe32pX2qXEIIga7aMCHIwSqxoq5I43EE4yM8mlm0DOryX9nqd9YtOytcQwGMxzlQACwdGIOABlSpIA9K1POX0f/AL9t/hR5y+j/APftv8KAMq70GW5upJIdd1S0ilOZLeGSMofXBdGZP+AsPbFaVnaQWFlBaWkYjggjWONB/CoGAPyp/nL6P/37b/Cjzl9H/wC/bf4UAZr+H7V9Iu9OMk3k3czzOwYbgzvvOOMYz+lW7/T4tQFuJmdfs86TrsIGWU5APHSp/OX0f/v23+FHnL6P/wB+2/woAWSNJY2jlVXRwVZWGQwPUEVgjwn5UP2W11zVrewxtFmkkbKF/uh2jMgHbh+O2K3fOX0f/v23+FHnL6P/AN+2/wAKAKQ0W0VtM8oPGmmZFuitkY8sx4Ock/KfWpNT0uDVbdI5mkieJxJDNC+14nAI3KfoSMHIIJBBBqz5y+j/APftv8KPOX0f/v23+FD13BabGdY6K9tM0t7q2oakxQoBdNGqAHr8kaIpPuQT6Vmy+CIJ9Kk0qbWdWfTWhMMdqJ0VYlIwAGVA7ADoHZh6g10fnL6P/wB+2/wo85fR/wDv23+FAFDVNFj1JbZlurmzuLVi0NzbFd6ZGCMMrKQR1BBHTuBVy0ge2tY4ZbmW6dRgzTBQ7+52gD8gKf5y+j/9+2/wo85fR/8Av23+FAGDF4UmtXuPsHiTVrSGe4kuDBGlsVVpGLNgtCWxknvVseHLJdHTTkaYRrcJctIZN0kkiyiTczHOcsoz7cDHFafnL6P/AN+2/wAKPOX0f/v23+FGwbkV/Zx6jptzZTlljuYmicocMAwIOM9+arajosOo2MFu009u9s6yW9xCwEkTgEBhkEHgkEEEEE8Ve85fR/8Av23+FHnL6P8A9+2/woAyp/DxurO2SfVr9ru1kMkN+DEsykggjAQIRg4wUx+PNFj4cis9WTU5b++vbxYHgMlzKCGVmU/cUBVPyD7oHvmtXzl9H/79t/hR5y+j/wDftv8ACjrcDNfw/avpF3pxkm8m7meZ2DDcGd95xxjGf0q3f6fFqAtxMzr9nnSddhAyynIB46VP5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAo85fR/+/bf4UAPopnnL6P/AN+2/wAKPOX0f/v23+FAD6KZ5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAo85fR/+/bf4UAPopnnL6P/AN+2/wAKPOX0f/v23+FAD6KZ5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAo85fR/+/bf4UAPopnnL6P/AN+2/wAKPOX0f/v23+FAD6KZ5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAo85fR/+/bf4UAPopnnL6P/AN+2/wAKPOX0f/v23+FAD6KZ5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAo85fR/+/bf4UAPopnnL6P/AN+2/wAKPOX0f/v23+FAD6KZ5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAo85fR/+/bf4UAPopnnL6P/AN+2/wAKPOX0f/v23+FAD6KZ5y+j/wDftv8ACjzl9H/79t/hQA+imecvo/8A37b/AAprXEajLbgPdD/hQBLRUP2yD+//AOOmgXULHCsSfZTQBNRULXUS/eYj6qaT7ZB/f/8AHTQBPRUP2uH++f8Avk0C7hZSQxIBwSFPBoAmoqD7ZB/f/wDHTSG+t1GWkwMgZIPfgUAWKKg+2QDq/wD46aPtsH9//wAdNAE9FQfbYP7/AP46aPtsH9//AMdNAE9FQfbbf/np/wCOmj7ZB/f/APHTQBPRUH223/56foaBe25ziTocHg0AT0VB9sg/v/8AjppVuomOFYk+yn/CgCaiomuolOGLA+6n/Cm/bIP7/wD46aAJ6Kg+2Qf3/wDx00ou4ScByT7KaAJqzLL/AFMn/XxN/wCjGq/DcRXG7yXD7Thsdj6VQsv9TJ/18Tf+jGoAbdai2m6TZukImaTbGFL7QPkJznB/u1SPie4Xrp8f/gSf/iKdrX/IH03/AK6L/wCinrEkrlq1JRlZHpYTD06sLy7mqfFkw66fH/4EH/4imHxjKvXTk/8AAg//ABFYr1WkrneIqI9GOX0H0OgPjdx105f/AAI/+xph8eFeumj/AMCP/sa5l6ryVk8VVXU3jleGfR/edUfiDt66Z/5H/wDsajb4jqvXTD/3/wD/ALGuQkqrJU/W6vc3jlGFfR/edsfiZGvXS2/7/wD/ANjTD8UoV66U/wD3/H/xNcFJVaSl9cq9zeOS4R9H956Efivbr10qT/v8P8Kjb4vWq9dJm/7/AA/wrzeSqsneqWLq9zZZFg30f3npx+MlmvXSJ/8Av8P8KjPxpsh10e4/7+r/AIV5XJVR+9aLEVH1L/sHBdn9564fjfYDro9z/wB/V/wqM/HTTh10a6/7+rXjz9KrPWqrTYf2Dguz+89nPx500ddGu/8Av6tNPx90sf8AMFvP+/q14k9QPWyqSZLyLBLo/vPpnwP8SLXxxf3VtaWE1r9mjEjNK4OcnAAArpYuj/8AXR//AEI14x+z/wD8h7WP+vZP/QjXs8XR/wDro/8A6Ea3i7o+PzGhDD4qVKGyt+SHad/rr7/r4H/otKtQf8e8f+4P5VV07/XX3/XwP/RaVag/494/9wfyqjgH0VkCztrjUL57i3ilYTKAXQMceWnrUn9mWH/Plbf9+l/woA06KzP7MsP+fK2/79L/AIVGbO2t9QsXt7eKJjMwJRApx5b+lAGvRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUhXLBskY7djVW5uXjk2qBj370tuwfMhJHqCeBQBYLEg+XjPv0pwzjnr3pOi/Jj2paACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACq16cqijqTVmmtGr43DODkUAV47aIydyV65qcRomSqgE9TT8Y6CogXfO9duDxg0AVZ3JZklwBjK4qoPvVPcSiRgACNuetRAUAPXrUNrIPs/lllB82Qbd3Uh2GcfrUmQoJJwAMk+lVo7eKWaa4lt0/fYCiRQTt9x2z1x19fQAFh1phAPDAEdwRkGmQoEM23ds8zCLkkKAoBx+OaeaAIoML5sIUjyXIB24GD8wA7cA4x7VLUYzHNjOUlJIz/C+On0IB/KpKAAUhOKUDNBAoAbml3Z4ptLQA4LUa/JcGMfckVpAP7rAjP57s/XPrUimo9mL992SVhUAE/cyx7e/9KAJB1q3ZrmcH0FQRRl3Cjqa0IIPKkbGcEDrQAy+H7sHHfk1QrYYZWs+7lggtXuJkKrEpZgBz/wDXoAqnrT7cEzj25qA31zGyvcWAt7VpAhLyAuN3AYgZAGcDr3q0xS0Es0zbIo1LFj2FADdAO8XkqgiKScvGT/EuTz+eaksv9TJ/18Tf+jGpNAikh0yFJlKSCFNynseTS2X+pk/6+Jv/AEY1ADbrTm1LSbNEmELR7ZAxTcD8hGMZH96qR8MXDddQj/8AAY//ABdWdWju5fCbR6fHPLO8UahLaURyFSV3bWLLhtucHcPrTfDTWywXFvDBqdrNE4M0Gp3b3EiZHBDGSQbSB/C2OD3zUOEZPVGsK1SmrRdiofCcx66hH/4Dn/4umHwdK3XUU/8AAc//ABddTRU+xp9jX63XX2jkz4Ic9dRX/wAB/wD7KmHwGW66kP8AwH/+yrr6Kn6vS7FfXcSvts4w/D7d11P/AMgf/ZVG3w4Vuupn/vx/9lXb0UfV6X8pX9oYpfbZwp+GcbddUb/vx/8AZUw/C2Fuuqv/AN+B/wDFV3tFL6tS/lH/AGli/wDn4zz4/Ci3brqsn/fkf41G3whtW66tN/35H+Nei0U/q9LsV/amM/5+M82Pwbs266vP/wB+R/jUZ+C1keusXH/flf8AGvTagvbY3dq0PnzQBiMtA21iO4z1GfUYPoRT9jTXQf8AamN/5+M82PwQsD11i5/79L/jUZ+BenHrrN1/36Wuju7q7fw54ft0XULr7VEpnFnOEnlCxZ2iRnXBJ5J3A4B55rVttRSDwhc3NjHdB7OGX9zeyGSVHQE7XYsxY5HXccjoTTdKCvpsNZpjm0vaM4Q/AbTT11q7/wC/S00/ALSz/wAxq8/79LXV6NJHEs9vevrdhdT2pla7v7tXWRVxvkjAkdI8bgcYXgjjjhraffJo+oHSV1eW2n8kRQy3zm4cB/3ro8rho8oeAWXp0HevZxRP9qYx/wDLxkfgf4b2vge/urm0v5rr7TGI2WVAMYOQQRXSxdH/AOuj/wDoRqp4aa2FrcQQQalbSxS/voNSunuJUJUEfO0jjaRj7rEde+atxdH/AOuj/wDoRq7JbHFUqzrTc6ju2O07/XX3/XwP/RaVag/494/9wfyqrp3+uvv+vgf+i0q1B/x7x/7g/lQZlGD/AI/b/wD67j/0UlZmtqLvVtM065dls7jzWkVWK+aygFYyR2ILNjvt9K04P+P2/wD+u4/9FJWfd+HYNT1R7rVpDdRKoFtbkFVg45bg8uT/ABcYHA9SAVhZW2i+ItOi0hFt0u/MWe2j4QqqEiTb0BDbRkdd3Patmf8A4/bD/ruf/RT1DYaPY6bI8lpCwlkADSySNI5HpuYk49s1NP8A8fth/wBdz/6KegDRooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooApTxiZ2aNtzKcFaaiTQZY5YE/d9BUotmS43pjb3GamfdujA6Z5oAWJzIgYrgdven0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFACPnY2OuOKqpd7YiZDls8VPcNtt3OccVlmgCWdhIQ64x+tR02lB560ARNi4lKEEwxn589HbsPoO/rx71KMs/qc1FbncskhBDvId+QRjHAH/fIFOkTzIZIwQN6Fcsu4DIx07/AEoAjtV2pLlt5MzEvnO7pz+WB9QalKkDJBqFbZ0hEb3DlVUKFiHljGPXlv8Ax6kaFIVMsClWT5m+c/OB1BycE4zye9AEjoJI2RiQD0IPKnqCPcHmmEXRk+/AqBewJ3N7g9B16Gpcg4KnIIyDSUARNcxoP326Fs42sp5Popx834c+1JLOy20ksaZ8v74fgp6kjvgHOO4qXcR3pjxLKwLM64xna2NwBzgj/J96AEYNCwZ5XkTO19yqNvo3AB68fj7VIDkAgggjIIOQRS5JfOec5pI0kSRk8pnjaQlHT+HJB+YfUnpmgCe2tzK3PCjrV57KJm3ABSQASBycdKkhhESgCpKAGJEiKAFHAxmn0UUAFZF3Z3V5dGO4MaWiuHCISzSYORk8ADI5HP1rXpCMigDHuys9jeLIMxmJuD9DVKS7S90+ytI5VnuJmhaQKwJCqVZy3pwCPqa0C/kzPs6ZPFNiNvDvaG3SJ3HJRQM+mfzNAF+1OZpfYKP51Vsv9TJ/18Tf+jGqbTjkyk+39ahsv9TJ/wBfE3/oxqAJjZQaho0VtdoWjaNM7XKMCACCGBBBB6EHIpdN0m00mF47JJP3jb5HmmeaRzjGWdyWPAA5NVv9B/6d/wDx2j/Qf+nf/wAdoA1aKyv9B/6d/wDx2j/Qf+nf/wAdoA1aKyv9B/6d/wDx2j/Qf+nf/wAdoA1aKyv9B/6d/wDx2j/Qf+nf/wAdoA1aKyv9B/6d/wDx2j/Qf+nf/wAdoA1aKyv9B/6d/wDx2j/Qf+nf/wAdoA1arX9hDqVqbe4e4RCQSbe5kgfj/ajYN+tU/wDQf+nf/wAdo/0H/p3/APHaAGReFdLh0saei3Zt1cPHvv52kiIGBskLlk44+UjqfU1Pa6NDZSwi2eRLaGJ0FuXZg7OQWdySS7cdTk8sec1H/oP/AE7/APjtH+g/9O//AI7QBGnhPR44LiFYJ/LuIzEwa7mOxM52Rkt+7XgcJtHA9BTovDGnR200Ba/mSbbu+0alcSlSDkFGeQlD7qQad/oP/Tv/AOO0f6D/ANO//jtAFnTdKtNJgeKyRwJH3u8sryvI2AMs7ksxwAOT0AFMi6P/ANdH/wDQjUP+g/8ATv8A+O1PHs8seTt2dtvSgBdO/wBdff8AXwP/AEWlWoP+PeP/AHB/Kqunf66+/wCvgf8AotKtQf8AHvH/ALg/lQBUayuRczyW9zEqzOHKvCWIO0L1DD+7R9lv/wDn7tv/AAGb/wCLpz6iVuJYo7SeXymCsyFAM4B7sD0IpP7Qk/6B11/31F/8XQAn2W//AOfu2/8AAZv/AIuhbK5NzBJcXMTLC5cKkJUk7SvUsf71L/aEn/QOuv8AvqL/AOLpU1EtcRRSWk8XmsVVnKEZwT2YnoDQBcooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKCcDJ4FABRUcU6zFgn8NSUAFFFFACM6oMuQB70BgRkEYqjeuTIF7AZFQRytGflP4UAa1FZiXUi5z82fXtSPdSuOWwPQUAalM86PdjeufrWWZXK43sR9aZ9aANFr1Vcqy9O4NTRSCWMMowD61k1qw7fKGz7vagCSiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAZMnmRMvftWUwOa2KrT2ocl0OG6ketAGfTZS6Qu0ahnAyAeakYHNNoAggYO0j7nMhKhwYygHHHH0PqT0qYGolJ+2TqwIIWMrz1XB5/PcPwFSUAOY5qCcbgsbkrHLlHYLnrgBfbOTzUtRSkCSMvu8sAnhSRuBGM4GemT+FAEpO4knvSHpTI5VlyIyxwM8oRx68jmnUAJRRTgpJGKAHRxs7hV6mtWCERRgdaisoPLUsw+Y/pVqgAooooAKKKKACiiigDNvUCTcdG5qtVzUB86nPbpVOgC7p3/AC0/D+tRWX+pk/6+Jv8A0Y1S6b/y1/D+tRWX+pk/6+Jv/RjUAWLT/jzh/wCua/yqWorX/jyh/wCua/yrm9Mvr9LpE1XUrmK9dG8yyubVFiLbSf3EiqN2MZwWc7eoBpN2QzqaKwY9Su20DQLky/vbtrcTttHz7ky3GOMn0rK0/wAR6lJ4VvftEqvqqzGK2fYAGEjHym2jjC8g/wDXNjTl7t/L/gf5iWtvP/g/5HZ0VzWmatfX9pB++wbfTIrm4k2DM0siEgegA2ljgdSvYEGvpevahJpmlwahKv8AaJlg891QAXEUikhwO2cYOOhU9sU7e9y+i+92FfS/9aHW0VkyeI7SPT4Lxo5/LnWVlAUZAjVmOefRTiqjeLQryJ/Ympl44Bc7f3I3RHPz8yYHTocN7VN0M6GiubHiV4L7UHltbq4sITG/nxLGFt42iViWywY9SeAxFWF8W6c+tf2cm5m8ww+askZXzBnK7d+/qCM7ce9PbcOlzcorL07XodQ0Y6q9tcWdn5InWS5Cjcm3duwGJGB649s1mav4pnt9Dv3g0y+tr1LV57dJRFl1GAXHzkYUspIbB5HBo62A6eiub1y9urLStIBvNRtWnuFjnkit457jHlO2NiI6k5UZ2qR1xxUFl4nWx0p7q+uZ723muPLsZJ1igmnATLZDeWq4YOPmCnjpnGQOx1dFYkfiiC5ghfT7G7vXkQyNFB5e6JQxUkkuFPzAgbS2cZGRzUkviOCK5Zfsd21vHIsU12FURxO2MKwLB/4lyQpAzyRg4ANeis/TdXXVJJxDZ3McUMskXnyhQrsjlGCjduPK9cY/lWhQAVWh+43/AF0f/wBCNWarQ/cb/ro//oRoAdp3+uvv+vgf+i0q1B/x7x/7g/lVXTv9dff9fA/9FpVqD/j3j/3B/KgCjB/x+3//AF3H/opKfdXMVlZzXVy+yGCNpJGwTtVRknA56CmQf8ft/wD9dx/6KSue8bWGrXGj6hcWWtfZLOOwk860+yI/nYVi3znlcjA46YzQB0lrcxXtnDdWz74Z41kjbBG5WGQcHnoaZP8A8fth/wBdz/6Keue8E2GrW+j6fcXutfa7OSwj8m0+yInk5VSvzjlsDI565zXQz/8AH7Yf9dz/AOinoA0aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKr3VwYdoXqfWgCSWZIRlvyqjcXTTfKOF9PWmTzGZ89uwpqLvcL6mgC3YKQrMeh6VcqONCuOeAMYxUlABQehoqrePJGAUOB9KAKc5Yztu9aZQWLHLHJPWigAoozRQAmKWkJxU1vCZnH93uaAGxwvKwwOPXFaiKEQKOwpVUKoAGAKWgAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCGe3EoyOG9fWs+eM26lpcKv96taggHqM0AYKAlpJCpTeRhT1wBjJ9M+lPq3dxRxZx95uQPSqlABTRnPHFGMUUANlVsF4y3mrggA/fAOSvpyM/iaiguPMkeKdoUmBBESuMgFc4I7kDripx1pLiNz5AwwYzLtfkbRyT+eMfjQA5Rk1pWtt5fzP1x09KS1tdg3OOfp0q30oAKKKKACiiigAooooAKKKKAKtzbeZmRTzjkVQIrXkbbGSfTsKyGJLE0AXNOGDL+H9ahsv8AUyf9fE3/AKMaptO/5afh/WobL/Uyf9fE3/oxqALFqM2UIPTy1/lWdb+H1guoJJdRvbqG1Ytb287IUiO0rnIUO2FYj5mbr681p2iE2UH/AFzX+VTbD7UAYlp4bitJrf8A068mtbVt1tZylPLhOCBghQxwCQNzHH4DBD4YsYbq0nV5y9rE8aAuMNuzywxyRucD03mtvYfajYfajcDEt9BFjNGLOQm3azSznWQ/MVQEI4IGN3JBGBnIPbBfL4csZjpTMZRJpRHkSKwBYBcbW45BwDj1ANbGw+1Gw+1F9bgc8PCVuV8uTUL6SBRKsULGMLEJFZWAwgJ4Y43En9avPotu9xJKXl3SWf2MgEY2ZJz068//AFq09h9qNh9qVkO/9f16GBJ4WhdplGoXqW9wqJPbKY9kqqoTBJTcMgYOCOtWYdDW3v3ngvrxIHkaY2asoi3t1P3d3JJON2MnpWtsPtRsPtTeu4ulihb6XbQaHHpRDS2qW4tyJDyyBdvOMdqonwzFLBcR3t/eXhmt2tleYx7oY2xkLtQdcDlsngc1u7D7UbD7UdbgVLqyju5rSWQsGtZvOTaRgnYyc+2HP6VQm8OwPuNvdXVpJ9pa6ikhKZhdl2ttDKRhssSCDyxPpja2H2o2H2oAxpNB3eU8Wp30NwiGOS4RkLzKTuw25CByTjaBjOBgcUkvh2GS6ZxeXaW8kiyzWgZTHK64wWJUv/CuQGAOOQcnO1sPtRsPtQBRt9OW0tTBbTzRg3DzlvlJJaQuy8jGCSR64755qWO3kSRWa7mkALkqwTDbjkA4UH5eg9uuTzVnYfajYfagBlVofuN/10f/ANCNXNh9qpw8K3/XR/8A0I0AO07/AF19/wBfA/8ARaVag/494/8AcH8qq6d/rr7/AK+B/wCi0q1B/wAe8f8AuD+VAFGD/j9v/wDruP8A0UlWKrwf8ft//wBdx/6KSrFABVef/j9sP+u5/wDRT1YqvP8A8fth/wBdz/6KegDRooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqlfYyMj5ux9qu1HOyqnzY+h70AZXYnBOB0HepYFYzIUB7Ej0p0UPnuSvygEcVeigSIkqOTQBJRRSMwVSx6CgBaoXfmjIY5TPXFTPdAbSOUbqcdKryXbMSAo2+9AFbOKM0Y5oIoAQnNFFSRQmRvlI/GgCS2tjK2XB2/zrRRAi4UYoRlKgIQcehp1ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUARS26SsCw5qvPZDZmIHI7etXaKAM4WEhXOefSojZyhgNvWtaigCpHYIEG/Oe5q0FAAGOlLRQAUUUUAFFFFABRRRQAUUUUAFFFFABWZPGRcMqjqcitOmSYCl9uSB170AQ2x/fzewX+tVbL/AFMn/XxN/wCjGqewJZ5Sepwf51BZf6mT/r4m/wDRjUAXbP8A48YP+ua/yqaobP8A48YP+ua/yqagClp2rW2qed9l3/umw25cZ9CPyq7TI4Yod3lRom47m2qBk+pp9ABRRRQAUUUUAY/ijV20XSFuEu7OyaSeOIXF8MwxBm5ZhuXPGf4hzVnRbiW70uOebUbLUvMJK3NjGUidfYb3/PdUuoQ3s1sBpt5HaTq2Q8sHmow9GXKkj6MDx+FQaLpb6XbzC4nSe4uZmnmeOLykLEAfKmTgYA6knqSeaANGiiigAooooAKKKKACiiigAqhF0f8A66P/AOhGr9UIuj/9dH/9CNADtO/119/18D/0WlWoP+PeP/cH8qq6d/rr7/r4H/otKtQf8e8f+4P5UAUYP+P2/wD+u4/9FJWbrH/CT/bE/wCEf/sj7N5Y3fbfN378nONvGMY/WtKD/j9v/wDruP8A0UlWKAMXR/8AhJ/tj/8ACQf2R9m8s7fsXm79+RjO7jGM/pWlP/x+2H/Xc/8Aop6sVXn/AOP2w/67n/0U9AGjRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAASB1qKSNLhAfyNU7iGRGJySp5+lWbMYhB/rQBOihFCqMAUvSgnFV3mWSGRQcEAg0ATqwYZUgj2pJBuiYHoRVGyciQr261cmkVI2LDIxzQBls2PkDZUHIpKdJt8wlOmeKYTigBaM0hoUbmAFAAetSwW7SknOFHes7+1rf7R5extmceZn9cVqwx7Myv0XoPU1MZxlsaVKU6dudWuWraDyky33j+lSs6opZjwKoyXzFNqjHvmqpcnqc/WqMzUW7iY4yR7kU9po16uPzrGfzTJGY5AqL98etPzkn0pJlNWSdzWWVG+66n8acXUdSBWQDig9KZJq+fF/z0X86fWNV+ykLKVY529KALVFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRUc0whTJ5PYUASUx5NrBV5Y9BVVr1yPlUD3osTukcsct6mgC6OnPWiiigApGztOBk0tFAFa1Vkml3Ltzg4H41Vsv9TJ/18Tf+jGrQH/Hw/wDuL/M1n2X+pk/6+Jv/AEY1AF2z/wCPGD/rmv8AKqWq6wNPZIIY/OuXGQmcBR6k1ds/+PGD/rmv8q5vU1MXiSfzePOjQxE9wBgj86AJk8RXsDbr+2haH+IwE5UeuD1roo5EmiWSJgyOAykdxXIzSJHCzyfdA5rf0CKSHQbVJsh9mcHsCSR+hFAGjVTUdW07R4Fm1bULWxidtqvdTLGrH0BYjmrdc34quUsbrT7ttb0/SHh8zY+pQF4nyAMbvMQBvTk5yeOKAN+3ure8tUurSeKe3kXcksThkZfUEcEVXtdZ0y+tZbqy1G0ubeEkSzQzq6RkcncQcDHvWPc6jJq/gebeire3NlJJ9mjJ3yIOCyqfmwRjH+8B1rIvrzS9Tmub3SGiudMhhtVvHtxuQqk4YqccEqm7cvUA4NAHXx61pUultqcWp2b2CglrtbhTEMcH584/WrFtdW97ax3NnPHcQSDcksThlceoI4NcVLfabPqUmsQywyaIt/C810hBhLrE6mTd0KhjEC3QFfbjf8MSQz29/c2O02VxevJbMo+V1KqGZfUFw5yOuc96ANuiiigAooooAKKKKACiiigAqhF0f/ro/wD6Eav1Qi6P/wBdH/8AQjQA7Tv9dff9fA/9FpVqD/j3j/3B/Kqunf66+/6+B/6LSrUH/HvH/uD+VAGLdyyx3c6wyGIyXgUsoBOPIB7g+grD1rxBfaRqVlbRpfXwuI5ZH+zeSZI1j2chCoL/AH+gOeOAc1r6iXFzMYVVpBefKrNtBP2bgE4OB74NedT+Ibi1m1q51W4is/EMUCW0aGP91bRFgzGNtxEjbW344ZtijbxQB3mm6tHq8DS6fq0soRiki7EV42HVWUoCp9iAavWkssl3As0hlMd4VDMADjyCewHqa4+28UeFvCunXmmoxtP7HCRPA0W2SckDayj+Mse/c8njmur0yUTzwSqrKJLzcA6lWGbbPIPIPtQB0VFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAVbrJkRVbGRyM8VZVQqgDiqd1l7pEBx71doAhug/lZjJyDzis9kfG9gcE9a1qhuoy8B29Rzgd6AKVswSXJ/umrci+bZ56kjNZ+KsQ3WyPa+T6UAVzGyAFlIz0zSU+SRpWy34D0pg60AFIG2sCKc3SmUAU/7Jt/tHmb22Zz5eP0zV5nLd+PSkAJ6Uu3FTGMY7GlSrOpbnd7DaKKKozCjNFFADqR3RMeY6rnpk9aWqd9YvdyK8bgYGCD2qJtpXSNaUYSlabsi7ir1nCUUu38XQVTtlRGjheRSVHIJ5P4VrVSM2rMKKKKYgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArNnl82Unt0FWbqcKhRTyev0qjQAVLbZFwm31qKpIM+eu3rmgDTooqDE5n+Y4T2oAnooooAYP+Ph/9xf5ms+y/wBTJ/18Tf8Aoxq0B/x8P/uL/M1n2X+pk/6+Jv8A0Y1AF2z/AOPGD/rmv8qoarNpNw4s9QbdIPmARWLJ75A4q0s32fRRPjPl2+/Hrhc1y1mpMAmkO6Wb947HqSeaAL6Wmh2jiaeee4CHIEwZwv4AfzroYJ4rmFZoHDxuMqw71y9WfD0ph1O5s1/1ToJlX+6c4P50AdHRVHWJr6DTZZdMFv5qKWLXG4qoAJztGNx4HGR9ao3Wo6pcf2fa6SbSG5urc3Ek1zE0iIq7eAqspJJcd+AD1oA11tYVvHugg890EbP32gkge3JNS1jprj/8Im+rSQDzo4mLwq+R5ikqVB9NwxmqEms61p3nWeoixuL6RImtXgjaOPLyCPDKWYnaSDkEZB6CgDp6K5k6xrMTvpcpsX1Q3CRx3CxMsOxkZ95j3luAjjbv5IHIzxqaLe3V3BcRaiIftdpOYJWgBCOcBgwBJIyrA4JOOeTQBpUUUUAFFFFABRRRQAUUUUAFUIuj/wDXR/8A0I1fqhF0f/ro/wD6EaAHad/rr7/r4H/otKtQf8e8f+4P5VV07/XX3/XwP/RaVag/494/9wfyoAwdQdY7uV5GCqt8MsxwB/o4rLhsNEjt7yCZ7a6jvZ2uJxcMjh2JGMg8YACgegUfWukiOLy/IBP78cD/AK5pUvmN/wA8n/Mf40AeW654Ql1vXLnW5fEFrFqFsy/2RsC+XAq84kBzuJOc+nUeg7rSWZ5LdpJI5XN38zxD5Wb7PgkcnjOe9bHmN/zyf8x/jUUpzeWBII/fng/9c3oA0aKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoopDnacdccUARSweZKjqcFT+dTUyJCiYZix9TT6ACg4A56UU2RPMQqSRn0oAaYo3BOAcjB96qmPzbknbhF46dasxJ5KhDkn1xTbpmEOEByfSgCjO4eT5RgAYpgOKGYscscmkoACaQDJpc0A0AOGBQzUhGaQL60ACruNPwBRgAcUxiaABjTaKKAFzSg02lAoAij0ySXUftCPlN273HtW8BhQPQVHbIEt1x3GTUtTGKjsaTqSnbm6BRRRVGYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAIzBVJPQVUN/8AMcLx60y8mJfyx0HWmRRoy5dwoB6etACm7lZvlOB6AUNdTEYyB+FXII0CBlTaT69aq3ZHnYAxjqfWgCJYpHOdpOecnvQ8ZTGSMnsD0pRM6oVB4PH0qOgAq9ZJiMse54qko3MB6nFaqqFUBRgCgBaKKKACiiigBg/4+H/3F/maz7L/AFMn/XxN/wCjGrQH/Hw/+4v8zWfZf6mT/r4m/wDRjUAWoI1m0uOJ+VeEKfoVrlVVtPlNld/I8fCMeBIvYitu/wBQew0O3+z48+ZUjjz2JHX8KqJqd6qANcbiBySi8/pQBTkmjiTdI4Ue5rS8O2khkm1CZCnnKEiVhzsHf8TUf9qXn/Pb/wAdH+FMj1m6h1GA3Eu+2lby2UqBtJ6HIFAGtrFrf3unvb6Zd29rJICrvcWzTDaQQQAHTB98n6VmDQ9YW2s3TV7RNQtEaFZlsG8p4mC/KYzKTuyoO4N+FblzdW9lbvcXk8dvCgy8krhVUe5PAqO81Ox0+y+2X97b2trx+/mlVE56fMTjmgDNj0ab7A+iyNnTvshja4J/fSSsTlhjgY69OrDHSof+EbvbmK4k1TVlnvXWNYJ4Lbylh8tt6nYWbcS2CeQDjAAregniubdJ7aVJYZFDJJGwZWB6EEdRVO313SbxLl7TVLKdLT/j4aK4RhD/AL+D8vQ9fSgDN/4R2+eF7ibVI21YzrOlytqViTapUL5W8nbtZsjfnLE5HGNPSdPk0+1kFzcfarmeUzTzBNgZjgcLk4AAAAyeB1NMXxDor6W+pJq9g1hG217oXKGJTnGC+cA8jv3qxYalY6rai50u8t723JKiW3lWRSR1GVJFAFmiiigAooooAKKKKACiiigAqhF0f/ro/wD6Eav1Qi6P/wBdH/8AQjQA7Tv9dff9fA/9FpVqD/j3j/3B/Kqunf66+/6+B/6LSrUH/HvH/uD+VAFGD/j9v/8AruP/AEUlWKrwf8ft/wD9dx/6KSuP+K13cWPhmyubOZ4Zo9QQq6HBB8uSgDuKrz/8fth/13P/AKKesvwhqeqat4dhutas/s07dD081ez7f4c+n49DWpP/AMfth/13P/op6ANGiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooqOPzCSz8DstACyypEAXOM9KqzXgK/uz83rii4ul37dm7ae/rVebBIKgBSOMCgCOinIhdwo6mp5tsKCOMcsOTQBV4pRTgKQ9eKAFyKaTSbfWlI4oATdSE5ooAzQAUYp2KVcZ5oAQCrNtb+adx4UH86pwmdppFljCoPukVsW6eXCoIweppJ3RUouLsSAYGBRRRTJCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoopksoiTcRnnFAGe6YuCsnGTyRSPgvhEK9sd6tyRpcEOknzbcAZ4/L8P50SgxQhyAz9C2OlADDeYZdo+XvmornmbdnIYZFMjCtIA5wO9F0VnjAieSFgchkA9Mc+o9qAGUmDVUrc2oaUStd72G6MgjBLY+TrgYPT2z61Ib2Aj91J5rnpHHy59sdvxxigCwpwwPoc1rAggEcg1iJ5ry75N0aKCFjyMk/wB5sfoP8jXgOYEx6UASUUUUAFFFFADB/wAfD/7i/wAzWfZf6mT/AK+Jv/RjVoD/AI+H/wBxf5ms+y/1Mn/XxN/6MagDM1oERaPIfuAFT9Sox/I1DWtcRW99o6Wk/mowRcMImJRgOvSqyafGEAe6kZsckWzDNAFKqt+C9sI0++7qqfXNbH2CH/n4k/8AAZqIdOt01CK5lnkkWLlI/s7D5vUmgCfxNd6VZaLLNrVxZ26AN5L3bqoEm042lv4sZ6c9a5q81a2nh0HU7DWtJis4IJIze3I8+3im2pgErIoVtu4Alu5Heu1+2w/9NP8Av03+FH22H/pp/wB+m/woA5+bUJtW8DXCx+UmoXFnK0cUJIMigld6L97a3BH+8OazLTxNZQafdLf3Wi6lptrBGRJaQ+TFA+8KkMm93CnJBySNuDkCutV7Nbx7oI/nugjZ/KfO0EkDpxyTU322H/pp/wB+m/woA42a90u802bUrfxPoy3Bu4ZZ7mPE9tCUyUR9rrjp95mGT6cKOi8NarNq+mvPPJbzhJmjS6tVKw3KjHzoCTxyR1IyDgmtD7bD/wBNP+/Tf4UfbYf+mn/fpv8ACgCeioPtsP8A00/79N/hR9th/wCmn/fpv8KAJ6Kg+2w/9NP+/Tf4UfbYf+mn/fpv8KAJ6Kg+2w/9NP8Av03+FH22H/pp/wB+m/woAnoqD7bD/wBNP+/Tf4UfbYf+mn/fpv8ACgCeqEXR/wDro/8A6EasfbYf+mn/AH6b/Cq8PKMcEZdyMjHBY0AO07/XX3/XwP8A0WlWoP8Aj3j/ANwfyqrp3+uvv+vgf+i0q1B/x7x/7g/lQBRg/wCP2/8A+u4/9FJSXunWmorCL63SdYJRNGrjIVwCAcd+pqZ9OLXEssd3PF5rBmVAhGcAd1J6AUn9nyf9BG6/75i/+IoAkqvP/wAfth/13P8A6KepP7Pk/wCgjdf98xf/ABFKmnFbiKWS7nl8piyq4QDOCOyg9CaALlFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFIwJUgHBI4NLRQAyKMxryxY9yTT6KQ5xx1oAqXcKAbzhfoOtVRIyjap+X0Iq08M8ow5Xr1xSrYrt+YnPrQAQbBDv2hc9TVWaQPMSDkdqs3QMcG1Bhe59KoCgBxOaKTNLQApNJn1oPSgDI9aAA89Kq3l4bNkVYwxYZJb+VWR0pTggblVsdNwzipkm1ZM0pyjGV5q6FHzhSARuAOPSmyJcrdoF2iLHzA0rMFwZHVN3Tc2M07JHBoavoJPld7F+3t1UBydx7e1WKqW1w7uEYZGOoHSrdUQFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVXvQTBx2bmrFcx/bmsSw3lzHaWk1rbXEkRiDssrBDyR2zigC/zU7XBaDyyO2M5rNk1axSOGQzHbNGJkARmOz+8QoOBTp9UsbcxrNcqGlj8yNVVmLr6jAOaALQozVJtVsEtoLg3S+VcZ8khWO/HUAAZz7UyPXNMdkC3ane+wfI3ytnGG4+U59aANCjJNVbjU7K1nMVxP5bKQGJRtqk9MtjA6+tOl1CyguhbzXKJL5ixmPBzuYZA6en5UAWKcszxjCNiqGn6itxZQSXToks80kSKAfmKuw4A9hya0HhZIw7cZ7UAHny5yXNaMTmSJWIwSKzYkMr7R+J9K1EXagUdhQAtFFFADB/x8P/ALi/zNZ9l/qZP+vib/0Y1aA/4+H/ANxf5ms+y/1Mn/XxN/6MagCxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA3Tv9dff9fA/9FpVqD/j3j/3B/Kqunf66+/6+B/6LSrUH/HvH/uD+VAD6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAE3Ddt70MwRSzHAFIWQN1GT+tVXP2m5CqcoOaAC5YNa5zjOCM96zpH8qB5du7YM49at3xPmhT0A4pkiiOJCo+8Oc96T2HFpO7Kdpcm6hdmQIVOOOhqcUsa5ZUVQq56KMU3z0a8a3CMMfxVK91JSZpJc8m4KyDFQXv2rMf2Xdtxzs9fequoa2tiGAg3ODhQTgfU0y313E6RX8aw+YPleMnA9iKylVpv3bnXSwteC9oo3NjD+WDtDPgbhnAz3pCMMcVQuQ2oMhspAyoMEbsYPrThrcaHymj3hRgyccn1qvaJPXYy+rTlH3dX1XYq6vbzNeBwCyMAF9vat/SrYpYItwAzAY9cV59d3c13dNJI7Ek8DPSt3Q9SvI7WWLa7lCMZzkCuWlWj7RvuenicJU+rxjdaHZIixjCACnVlJM6lXyQ2MkGrsN4j8PhT9etegeDsWKKRWDDKkEe1LQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFcjDHq9tb39lb6XI0k93NJHO8iiMBjwTzn3xiuuooA4saXcaZeL5X2uWD7LHATbFNwZMjkN2Oc8Vb03T3tNStGWJ44IbAwje6sVPmAhSR3x6VtXibZs/3hVfNAGNYafcwnSfNj2/Z5bppPmHyht2388ilksLk6XfQiLLyaiZkXI5TzFOfyBrXKt6HmoWuY037g4RCQZAuVBHXkdMepoAydXg1G9j1G3aK6lL5Ft5cirDswMZ5yWzmtGztGh13ULuSMDzDEI5MjkBAGx6cire3nBpaAMTRtMutNvYLqSNpxI8iSRMwJgVpCQy+x4yOtdbLCJCCcnBHGeKoRf65P94VqHgcDPtQAxYwsjMOpp9FFABRRUZuI1cqTyO2KAFH/AB8P/uL/ADNZ9l/qZP8Ar4m/9GNV9f8Aj4f/AHF/maoWX+pk/wCvib/0Y1AFiiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBunf66+/wCvgf8AotKtQf8AHvH/ALg/lVXTv9dff9fA/wDRaVag/wCPeP8A3B/KgB9FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFQXUjRxjZ1JwKnprRh2BPVelAGWxcy/O3zdM56VoW0PlIc4JJ6iqz2zyzMwG1c9+9XwMACgChqAO9T2otkE0OJFyFPBpt65MwTPAp9xK0JVIztAFAD4owkjKVGAcqcVVZSJWwOAcZrSjO+JSR1FYV9Y3bagHRsIMYbP3amTaWiuaU4Kbs3YytUtH85zOPlkJw+OK56OF5ZiN+QhxuzXeSywrJ5UjqC/wDAwzn61XubCGePy9qwlTkFFArjqYbmd4s9jD5i6cVGcfmU9DTDSy7ugxirB0m3+0eZvbZnPl4/TNTWlolnEVRixY5LGp63hSXIlJHBWxMvaylTejMmbQI2vPPt5AnOSjDir9nam0R8sGdiPu9hU9CfMwBqo0oRd0jOeKq1IckndCHJpQKq218bi7MJiCjnBHUY9asTym3t2lWPzNuCQW28Z5OcHoOauMlJXRlUpypu0iaK6MNykSbWZxuYE42rnGf8PXmtWsSFGiVzLtM0jbpSvc9hnuAMD8K0La6G0JIcHsTVGZbooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKrX7Trb5tmRXz1fOP0qzVW+fEYX+9QBmT3N5tdntd78iNY5A3OD97gYHvTWjuxF8kyPIVIO8bVU44YYGTz2PX2qxgjgjBooArfYUQkW0ht1ZVVwi8sFP97rk5Iz1qyqqihVUBQMBccYpCeaBnNAFVJ0smFtcvtUlmilc/LtyPlJzwQWwPYCroO1uRnHY00gMpVwGVhgg9CKuxQRSrvwcH3oAlijTCyBQCR2qRs7Dt644pCyxqMkKO2TSq6uMoQR7UAJGCI1DZJxzmnUVBcxeYm5R869DQA+aURRk5GewqikmZ97kL68VGzEnJ5NIDmgDSRt1xJjsqj+dUbL/AFMn/XxN/wCjGqxZMXaQscnCj+dV7L/Uyf8AXxN/6MagCxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA3Tv8AXX3/AF8D/wBFpVqD/j3j/wBwfyqrp3+uvv8Ar4H/AKLSrUH/AB7x/wC4P5UAPooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKTcM4yM0ALRVSe7Ax5TZOealhuUkUZIDdxQBNRRRQAUUUjglCFOCRwaAM9082+PPGeamu7ZpACnJH60kNm8cgZiOPSrlAEVsrJAquMEVT1XURZqF2bi3GK0ay78RzzFZYwyqeKmV7aF03FSTmtCi9rDesl1uZQw5XHXFWXOSTTeAAFG1QMADtRmiMUh1Kjnp0Ww4dKQnmlpD1qjMWkHXikOe9KOtADhgEkKoY9SByaRnVI2eQqsY+8WPFRySiJNx5JYKq5xuY9BmjyiZFM8hkKNvCbV2q3btnj60BuNtSTAFKFVRiqEjG5B0ODyPT8KmFBPPPU1BNe21s4WaUBvQc4pNpK7KjCU3aKubdsxa3XP0qWqthMktsDGwZexHerVMTTTswooooEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUySJZVw4zT6KAKlxCGkAU/ORk571VZSjYYYNajIGZSRyvSql8BuQ98GgCpRRRQA5EMjBV6mpjLLbsUzkY4yO1Mt3CTqTgDpk1YvUBQP3HFAFRmZ2yxJPvUlvIImZiT0xgd6hooA0LVi6uScgtwPSpzyKz7RiLgAdG61oHkHBx70AZTrsdlPY4poNPmRkkYPyc5z60wDBoAt2H3pPw/rUFl/qZP+vib/ANGNU9h96T6D+tQWX+pk/wCvib/0Y1AFiiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBunf66+/wCvgf8AotKtQf8AHvH/ALg/lVXTv9dff9fA/wDRaVag/wCPeP8A3B/KgB9FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFQTXSwvtIJPtUkjFUJGBx3rLdi7lmOSaAJ2vGYtxgEYA9KZBIVmGBnPBqGpISRKCOtADjBhm5wB0JqVYFhj8x+WAzis0QxajqNy9yrS29uVjjRj8hYcsdvfnHX04q28zvkFjj0oAmN+5XAUA+tQPK7tlmOaZRQBIJ5FxhjgVqKcqDWPWjaTB49pOXHXNAFiiiigCK4m8mLI6ngVlMxZiTyanuJjKQOmCarnigAooooAbJPFBtEz7S3QVJgcEHIPINVruxW7dX37CBg8dau21tvQIhwEAHPeoTlzO+xtNU1BOL16kZGaazpCu+Zti9M4zk+gHc1aa12ffkA+tVHtYxdPK+2Vg4aPPITCgfTPXn3qzEaEcyLJMFUqPkjBztJ7k+v8qeOtOPNJigBR94ZrmNShljvpTICdzZDeorp6zNdDNFD12ZOfrXPiI3hfsehl9Rwrcvcs+G50gttssqgs3yruGa6MEEZHIrzpVw42122j3IuLFPm3MBg/Wpw9S65exePw/JL2ie5fooorqPMCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAIpriOJljMirLICY1J5bHXA9siq12xYR564NN1WzN5AAjKssbq8bkZ2MCD+vQ+xqnaahBqFqshmiSZUzNGHBMZ7g+nPrQBLRTlVZAfKkSTHXac0FGHUUAWba3SRN7c89KtsoZSGGQaoI0ojwoAGMYpsU7wt14zyDQA+5gWIjZkk9qrZ9q1wQVz2xmo5YEkQ4UbscGgCpaAmYHHQVfU7lBFVlJgiEYGJG7ip8rDEN5wAMZoAHhjkbLrk1E9pGR8uVPtUqSpJ9xs08+9AFSyGJJR6Y/rVey/wBTJ/18Tf8AoxquQsjTyGPkYXP15qnZf6mT/r4m/wDRjUAWKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAG6d/rr7/r4H/otKtQf8e8f+4P5VV07/AF19/wBfA/8ARaVag/494/8AcH8qAH0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRTJkMkRVTgmgCGe6VV/dkMaoE5JJ71Y+wy5/h/OmtaugJYjA/WgCNYywySFUdzUN7e2+lW7zXEsfmBSUjZwpb6ZpmsWc1zbRNEglSElpLR32pMMdCfY888etQ6OtlcWbzxWBj+YfNKRJvI6bWJOVB6H8qALdvA9paFJWDSyyNJJt6Ak5wPalpzuXbJptABRRV20jDQneMjORmgClTo3Mbhl6in3IxMQAAAOwoiZEJZ13HsKANGJy8aseCRT6qx3m+QKygA96tUAUprYCcHOFY8+1E1jxmPj2q3Im9cZxzkGnUAZcCjztki5yMfSpJbNlYeWMjvk1dMSGQPt+Yd6fQBSgs85Mw47DNWljSJTsAFPqteSlIwo6tQBTlkZ5DySM8Uw571i6/rT6dtgtsecwyWP8IrEtfEt/DMGml85M/Mrelcs8VCEuVnpUctr1qftI2O0opsUqzwpLH9113CnZ4rq3POaadmJ3qhrd19m08KFDNK2ASM7cd/rWhVe9sk1C28mQlWByjAZwazqKTg1Hc2w8oQqxlPY5OOVzIoLcE4ru9GsxZW+wuGYnmuftNCS3uDLdXCOsXzFE/rVyLUFk1ASeayIB90965KMfZ6z3PVxk/rC5aWy1en3HTUVBa3cd0m6M5qeu/c8Rpp2YUUUUCCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAaUU5z361nXtlaySK89rDNg5VnQEg1oPGJFIcZGeKHiV4tnbHFAGK2nae+CtssDL0e3JjYe2VwaYYNStmeKxkWaGQDZJcSlmh657ZYdMZNWnUoxDdRSByBjJoArNp1xNIk8+oYuozmPy0xGvGCNuec56k9h0qRX1fK74rNgG2kB2HmDPUf3eO3NS0uT60AaoAUADgCkVNrEgnH93sKI23Rqx7gGnUAQbWe8yR8qDikvSPJAIJ54PpVigjPWgDKR2jbcpwale7d1AHy+pFWJrRX5TCtUa2Jx8z8+1ABYfek+g/rUFl/qZP8Ar4m/9GNVy3i8mWRQc/Kp/nVOy/1Mn/XxN/6MagCxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA3Tv9dff9fA/9FpVqD/j3j/3B/Kqunf66+/6+B/6LSrUH/HvH/uD+VAD6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKie4jRyrHkUAS0VS+3nJ+QEduad9vXH3Dn60AW6KjM6LEJD0IpFuYmGdwH1NACXVzHaW0k8pwsalj9BWReatdJZ/akslkQYJ/fY2j1PBPHOfTFXLoq7fu5TuORkc7fpVezkCxIoGFXKEZz90lf6UAVo9KW5tjJqU0krz5MggnkEZU9FAB5XHHv1q7Kw4RMbABgAVmxTnTZyluGuNNMiRZ3f6hiSuF45UHb34z36VqSQ4yydPSgCIAnoM0ksttbNEl3cRwvMcRq7AbjU1uDuJ7YrP0mP7dHcT6kFluSxt5omQbIwpztA7g5ByevFAF2RNjd8djUsd26KFIBGMVQ0lpHt7izn3h7SUxpv+8Y/wCAk45yP5VYIKnBGDQAMxZizdTRmkpQCxwBk+1AFlDusyT/AAng1eX7o+lUbaF2bDhgnUg96v0AFFFIzBVJPSgBaKz0nZ7xTyVzwK0KACqd9Gxw45AGD7VcpCAwIPQ0AcD4q0+V50vIlLptCtgfdxXP29rNdTLFDGzMTjgdK9IddjspHGcYNRqFRvkRV/3VxXFUwinPmuezh81lRoqm43tsMtYPstlDBnJjQAmpaKK7ErKyPIlJybk+oUqnDCkopklSHTfLuHeSQMhBGB1OfWmrpMSzbjISn93HNWrq7t7KAzXcqxR9MnufQDvVG11/TLycRRXBV2OFEild341g40ovlZ3xqYqcXOKdvQ0I1WFAkQ2qOatwXhVgshyvr6VVK0m05rfY4W23dm1nPSiq9m+6DBPK8VYoEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUnQkk0tVbiGWWQ4wF7UAQXC4Yl2BcnoPSoD7UFcZz1pKAHZApyBS3znA9u9RilzQBpQ3CSEqOMDvUqur/dYH6GsmlVijZUkH2oA1qiuHkSPdHjjrmoBekR8rlvX1pLeZ5JwHbIPOKALMLtJGC64PfipKKKAGD/j4f/cX+ZrPsv8AUyf9fE3/AKMatAf8fD/7i/zNZ9l/qZP+vib/ANGNQBYooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAbp3+uvv+vgf+i0q1B/x7x/7g/lVXTv8AXX3/AF8D/wBFpVqD/j3j/wBwfyoAfRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABVW5jAiZtvzE9atHODjr2qpJI8b4l+ZSOwoAo0U7Y2MhTj1xTaACRpioCSRqijLGQE7cfQj/AD+VR20rT2scrxtEzjOxu3P9ev4091WRSkgyrcEA4z+NReXOisscxYEZDONzrx0HY/U/rQBN0NV5JHtJI1VfNWaRikSr84JOSQScEZPfGM9ahtJZ4rTL27kMMwgOZcHaPlY4JHzZ+nP4z2UZWLz3hxdSpmXftDE4A25HQcDj8+aAGbFa0YoklzBIzM9uSMnLEkqfUH37cc0yG2MEpudIiZZRiOe1uJCMjPHJzjGSRjg5NTWbKbRI13BoVWNw0ZQg7R2NNnmjjvIX+bdE4R5FJ2oG/hbB7nHrjjpmgClq8niCWzkhS1jSOQjdJbtvZUIwVxwSw9R1yK0baaztYYbSyhe3U4Cq0DoM4zySOvH1zVgyOBtyf600qJRskG9TwQfSgCrq1tA9rNqDNNFNbRMxMMpQuFBIBx1GT/nmpJdStrOyR9SmjFzHDveIOu5mA5AGeueKgOow2UM1vcS/aJFQ7BIQfNBGVDEcAnpz1xmpNKsYbe3QvbQrLKwkOIwCmeQPXIyetAE0UwuoVlS2ngzn5Zk2t+VXrFDuZiOO1TrAAwbcenTtUwAA4GKACiiigAqteB2VVRSc9cCrNFAFa2tvL+d/venpVmiigAooooAhuIy8Z2Llzxn2qNbJAo3DJA5561aJAGTwK5vxX4ttdA0h5wBNMzeWkQPVj6+gpSkoq7NKVKdaahBXbLrqBIwTkA8Gm14vd+OvEN3OZP7QeAZ4jhAVV9vU/jmuh8K/EO5kvorHX3WWOVgiXO0KyE9N2OCPeueOJg3Y9utkOKpU+e6dui3PR6KXad2KyrvxFY229Yy9xIvQJwpPpmtpTjBXkzxadGpVdoK5g+MZZG1aOJifLSIFB25zk/59K5+rms+IZdUkUT2kURj4UqTuA9Ceh/KrOmWUE9qLoSCXB2tGVwUPvXi1P3lRuJ9rh08Ph4xqKzWh2ulyyTaRaSzZMjRKST396tVQ0eaSW1ZZOVQgKf6VZuruKzh8yc4B4AHU17MJLkTZ8dWpy9u4Ja36FhHMbhl6itONi8YYrtJ7Vz9nqdvfMUiyrjna3eugjO6NSO4zVRkpK6MZ05U5cs1ZjqKKKogKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDNuITG5ychu9QVaukk8053MvUcdKrHg0AJQOaKUHFAC0Ug60tACpGzthRk1et7by/mf73p6VUhkMUm4Y9DmpTeyYHyqDjnBzQBeqKecQqOMk9BUaXasg3EBj7cCqksjOxLHPYUAXLabzpZGxjAA/nVSy/1Mn/XxN/6Map7DrJ9B/WoLL/Uyf9fE3/oxqALFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADdO/119/18D/ANFpVqD/AI94/wDcH8qq6d/rr7/r4H/otKtQf8e8f+4P5UAPooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigApr7dpLgYHXIp1Vri4j2FM5JHagBsl5HtIQHOODiqNI5YLlE3n03AfzpsV1a5UTsyFztCspGG9CegP86AH1Debhp10UYKRCxyTgdOn40+aQxQO4G5gMIv95icKPxJFRtZszL9ou5ZCh4CogHbqMHJ468e2KAJIgqRoixLAdobyQRlc8n9c0+ojaxeUywxoj8ur9Dv/vE9etLBIZYvnyJUOyQFcYbAP075oAhukla6gNrOYZmBV/k3ZjHPOeOCeP8AeOKmW3T7OIGDOudzfMQWbduzkc5zzTUZEmmMjqsjMF+YgfL/AAgfr+OaS8jeZEt0Yr5jfvCP+ef8Qz2zkf55ABX+3SpJLHGiTx2rASuJdz7STxjsVwc59Klk+1y2bxS2Sl5lwFjmGFB7MT3HfAINWUVI0CRxokY6RquF/Kqslt5KwpazzQguiJEG3AkHPU5IAUEkAgHFAFmNFjiCBEUEfMEQAMe/FQxzC2kS1dCoZm8pwV2leoGM5GMhenpVk9TjpUcrxpEfNG5T8oXZuLE9gKANOxnSaFgkqyFGKnawOD6VZrN02UglNixQ9Ik2bTge3b2GK0qACiiigAooooAKKKKACiiigCC9R5LR1iOGIODXl/i6WPT9Hmgv0IknUrEhH3iO4+mQa9VdwiFj0FcJ4rjs/E1qbOYeS6PmGcD7jdOR3B71y4iKa31PXyqo4VtV7vXyPI9Osm1HUYbRGCGRsbj2HU/oK3de8Kx6Xp32u2ndwhAdXx34yMe9H/CG+INM1KMiJUKtuS4SQFT7jv8AgRXXrY/aZ7VtRlM0cLiRoVUKsjDpn2z2rg5XezPsa+MSlGdOSceyOlu2nTwyzOSJ/syhz3yQM/1ria6xdSNxM8Ny2Yn4YD+H3rN1TRmtLj5Y2aM9HUcH/CtcR+8tKOyPnsC1Qbp1NG9Tl9RXDI2OoIzW54EVn1C7DLui8n5gRxncMf1qO5037TGqOTFtOQNtaGjSx6FA6onmvLyxPFYUVy1FKWx34up7TCunDVs60AKu1AFHoBWXrtpJcW6SRDd5edwHp61bsb6O/hLoNrKcMp7VT1LUZre68qHCqo5yOtepUlB0/Jny+GhWjiLJaruc2rsjBkYqw6EHGK7fw9fyXNgnnnLZK59cVjRaHBdxrcb3iDclAM/lT4NUW2kS3hhCwIdo55+tc1FOk7yejPRxco4qHLTV5L8Dr6KZE++JW9RT69A8AKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACignHbNFABRVO4umDlY+NvBNQfapf736CgDTorJMrt1Yn8auWk+4eW2c9jQBaJwCazTcuJiysevT2q286yQyCP5nwQFzjNZUcvmtKCjRtE+wqxBPQHPH1oAtNdyN3x9BUZcucsST70yigB1IBTgMkCqiSXF2vyRvarub52KsWXkDA7c8+2KAJ/NjWYRF1EjdEzyeM9PpzT6qfZ4ba+tjArea5feWJfKbRkknPdUFWc0ARz4Fxallyu8jkZCkjg/XIwPrU5GDg1FIyLE3nYMeMMGXcCPp3plquy3AClFLuUQjG1dxwMduOcds0AT0hpc000AXNP6yfh/WobL/Uyf9fE3/oxqm0/rJ+H9ahsv9TJ/wBfE3/oxqALFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADdO/wBdff8AXwP/AEWlWoP+PeP/AHB/Kqunf66+/wCvgf8AotKtQf8AHvH/ALg/lQA+iiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACs2eBhKdqEA9AOa0qQgMMGgDHpcRSI0VygkicYKnmrsluUUrCmQ3Uk9Kr/ZZs42/rQBX8nDqzStMEHybgOPf3Pv8AWnVaFlJjkrVYjBIPUUAQ3VwlpZzXMoYpChdgvUgU1AEu55ThUmjjkLMw4YfKc+nG3/Oah1oE6DfgDJNu+APpVX7BFea85uIxIkdrAQjglWOW6/SgDR2oNUIYRmVoAoyQWwGJxjr3/SorWe2NzdxQ2ogMEoieRUGG+QNkkDgfNjmsDyybi4hvJreC9+2tIrtbu0zfPldrA8gjA+lXpnFrFrjNbxyRNexqRKhZFUpGNxA6gf0oA2hJH5fmeYpjBxuU7h+lVkkluZ42WIboy2ITJ9wkY3OQCAccAc/ernZY0+y60LZo5YXtoCDBAY0Y+YQcDJz9a076y+y6vcppMCwyvpcuzyl25fd8vTqaAL8moxR3dvblGaSUuHCMG8sqpboOvT2/pTp545U+zRjzJZfuo4ZMY/iyRwB6+vTmsayOmPqOj/2XEiuiyCUrEVYHyjwxxyc5rYEixatKZy3MKbHPKopJyDj7uSM5PXHtQBo2gImj3EsQOWx1OOtaVZNrqNqswXzo8sduC2Dn0rWoAKKKKACiikbO07cZ7ZoAje5iRsM3PtUoIYAjkGse8W5Rx5KruLZbPpVmG52HaSAnrilfWxTjZJl+isvUNTaC2MkIzzjmo4NWkbTWnkXG3g47/Sk5pOxapTcedLS9i7qVytrYySOMgDpXncRW5u2aXgEkgZrrnvo77T5zMG2r1Gc9emK5m4sIkhM6SMPmxsI/rXDiW5NNbHtZdFUlJT0exMzSBcEsyKPlyc4qvcSsm1o/lHc+9T2cqyTRwyg7WO0kUtzfrGz29rEhhDcl13Fvf2rn0avc71dT5UirYpJcX8apyzNzXZvJ5UbuxJCKWOO+BWXon2WW3aaGBY5lO1sEkfUZrSYqqMZcbMHdnpiu7Dw5YXvueLmFb2lZQtaxxn9oPd3TvMAGkJPHb2qVLeW4mCQqWY9AKzrs28d24tJS8Qb5WIwa2rDVtO0e0Es9wZ7iYdEH3R6V58LSlabPdrXp006Ubt7KxsaRYPYwuZvvv2HYVcmtoLhgZowxHQ1R03X7HVH8u3crL/ccYJ+laEkiRRtJKwREGWY9hXqwUOS0dUfMV3XVZuaakxwAUAKAAOABVX+zLX7R5u05znbnjNc9dePbWKYpbWzzID98nGa19G8QWetKRBmOZRlom6/hUKrRqPlvc2lhMZh4Oo4tJ/1qdHaS749p6r/KrFZtu+ydT68GtKug88KKKKACiiigAooooAKKKKACiiigAooooAKKKKACkZ1QfMfpTEnjkkeNHBdDhl7ipKAEVt3IBA9xik3EffAA+tMnnEK+rHoKz5JnlPzH8KAL093DDA8jOMIMnbyad9ojVRubmswc8N909c9MVWs9sdl5v+qif96Fc4ESkDA9vX6mgDYa+QfdUn9Kf5261aReDg/gazsHAPY9DnrUsUrplVG4HqCKAIiSevPvTallmjkvDbRxkSLGJCcADBJGPrxUVABVm1mjiPzjns1VqMUASTsGlcxkDPQ4zj8KhSPZuJd5GY5Znxk8Y6AAUSSJFG0krbUUZJwTj8BSxOk8fmQOsqZxuQ5FAC0CilHvQAA81ATNbzN5EPnwyy72CyAOhI5xu4IyM9eM4qY9aQnFAFecu6gywSRzod0Dwkt6fKcY9gQeD6+kiXGdouo/ssrfwuwKk/7LdD9OvtUoPpQwDoUcBlYYKsMg/hQBWhIvEEzSq8YbcscZ44Y7SxzyenHTirRz3/WongQgeWWgZVKhocKQDjjGMdvw7VGiLaSD55DFIAuZZWba4JOeemc/mB60AWKKUgjqMfWonmVGI2SSFRlvLTdtHv8A4DJoA0NP6yfh/WobL/Uyf9fE3/oxqfpUvnCZhHLGAcDzV2luvOOuPrTLL/Uyf9fE3/oxqALFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADdO/119/18D/ANFpVqD/AI94/wDcH8qq6d/rr7/r4H/otKtQf8e8f+4P5UAPooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAO9Rzy+TGW79qkqnf4wvPPYUAV5rqeSIrE4R+MHseenf6VAJQ6u7/u9jEPvI+U/X8c06ozBGZjKd5YsGILkrkAAHb07f5wKAFimV2PlSKxUZO08iiV3kbyFdx0Z2BI2r7H1OMfTJptyEMfmy7yYwduxyrEn+EEEdTgUW8PkiTdwZJNxG8vjgD7x5PSgBIs28qW0LnyihbYT9wDA4PfJOOfz4xSyu5uoFRt20kuu7gLjg/XIGPx+tVg8zXc8cSETyOFY5OI4l/izzyR0A78+tT2yOs90u8yJ5gIZvvZ2LxxxgDHP1+tAE0qmZVHmywlSSGiIBGQQeoI71Xht4rW+IgD/vo2dgZS2CGHzYJ6nd19qnlligUNPIsak4BY9fp61E7q9vLcQeaksSspyhByBuwR3H+JoAmknaPYMNIznCpuxnAySc9qjtomjjlMxR5J3LSbV4xgALz1AAx247VF9qFzKk1hE8x2ssuQUG0dFywHOTx261YSQSSOgV0dMbldeRnPfoenagB6HywoThV+6oHA/CtgcgVjqCzADqTWuoIUAnJxyaAFooooAKKKbJIIkLN0oAZcRo6EuOg61lnAwWZVBPG44zVpmluzhBhBVDUtLkmaNg4UKNpzUybS0RpTjGUrTdkSnjKsAQeoI60HBXYVUpjG3HGKMYVVBztAGT3opkXtsNSKJIzHHEoRuq+tUZxaTObARFAWwHB6NV8sERnbOEUsQMZOBnvxUMUZlSO5aKOG4dQ3zKW25HQjI5qJQurI3pVuVtyb/4JVtNDjtLgSyyeYVOVGKz9R0OWOSSe3KtDyxycFa27aORFbzBtBxtTzC+MDBOT6/55p8+z7LN53+r2Hd9KzlQg42N6WOrKrzN3voYmjXcdofIdOJWGXz0PbirviESDQ5xHnORux6Vxs+r5VhboVOeGJ6VqeH/ABBPNdLY6i3nRTfKC3UVx068XH2T6nsYjAzjNYmK21a72MGlezeZQRgfWuivLCC0vpo4ogq54zzxVCZAkhAGBXI6bi7M9SGIU0pR6mPbiaz1CErlXVxgjvzXZeNZpU8N/JkeY6h8fyqHRNHS6nW8uVykR+QHua37+yi1KxktbgfJIOvofWu6hRl7KXmePjcZT+tU218O549Wn4clkh8Q2hiJyZACB3FaN14I1SGYrbqk0eflYNj9K3fDXhNtMuBeX7K04HyIvIX3rlpYep7RaWPYxWYYX6vJqSd1sddBEzygqOFOSa0ajgKGIeX0HFRXNwY3Cp1717Z8IWaKofbJfb8qcLubuooAu0VDDcrLwflb09amoAKKKKACiiigAooooAKKKKACiikYhUJPQDNAFSGJbi4Fy6YZD8jDg49PcVcqt9tXywSMt3FQyyeVqFvcA4jmHlP7Hqv9R+IoAZeMTcHPbgVBVi7BM5IBx6+tQY5I7jqPSgBkiCWGSNjgOpUkdsjFQ5le/RLgRsqo0iMoIDMCo+6ScEZ9+ue1TyusChpjtDEKo7sfQDuaj2yyPG0iJGEbdw5LdMY6Y789aAAQPCzm1aNEc7jGycbu5yCMZ/GpoJRJyMqVbDqeqn0P+eaTNRzEQ/6V02YEnoUz3+nXP19aAHQ/Lrl9nljBEynuBlxj6ZBP408jFRWTefJcagwIWYCOEHvGucN+JJP0xUufSgBR0opOaDnFAAf59feqBt7WzuXaSFY4WUMkqAr5Zzhl3LyAcg+nWr4p2cUAVbdw0sgS6NxH5aMpLK3OWB5H0FT0qqqZ2Kq5OTtGMn1oNACUh6UtIfSgABpab0pQaAFNCnAweQeoPekd40XdI6xrnlnYAColufNybaI3EYwPMR1AJ74ycHHGeaAIRHd2yrBBI7QqFEbLGpdQMZUkkKOnUjvUlq8YdoWneS5f52WVNj4AA6YA/EVIZWPCW8u/0cAKPct0I+mTTVs4dmJI0Zy+9pEBRi2c5yOR+dAGlp/WT8P61DZf6mT/AK+Jv/RjU/S0kQzrI+9QV8ticnbjv75zTLL/AFMn/XxN/wCjGoAsUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAN07/XX3/XwP/RaVag/494/9wfyqrp3+uvv+vgf+i0q1B/x7x/7g/lQA+iiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiij6UANkkWJCzVmTzedJuxgYxVuZRGN8x3nPA7flVG6uY4lRxEdhcK5DD5AeAcd+cD8aAHRRGV9o4Hc0kqeUTv+VRk7m4GB3pyytHkJxn2pm47s5OfWgCnNcxymJInDIJVaVlz+7AOQD6ZI79qnS4gkfZHMjPjIUHqPUeo+lTAsxHJJ+tVVuYrq7i8mVZkjWRiycjd8oAz9GPSgCxk1VuYV88zymTyNg8wRybNpXPzHGCeD2PYcHtaowCCGGQeCD3oAjS2SKYSbpHlVPL3yOSduen6DnrUM0atfAzLKIZECbkYqruT0fByeAAM8c49KktmzE0eSTC7RnOc8Hjr14xTpwPs7liwCjfleuVO4fqKAC8HmWUoeQRqFyWb7oA5wfbjFQW7o8aJaCSFm+aYup3oB/CS3fPA68ZIqQA3TFZ0UxqgDqGOGcgEjHoB6+tWGJPU5oAWyhuVvVYTG4h2/N5oAZD6ggDIPv6Vr1VsAfLYkd+vrVqgAooooAKa6K+NwyPenUUANRFQYUYFRXiboD2281PUV0u63YegzQBlHrRn0o780tADZI1lhkifOyRSjYODgjFRwzTfa5opczQxJkzkBWDddpA6/KRyAKnXr9OeelUNIuftEMjeUFyxd2Bzuck7gew6cDJ4xQBT/ALem+0Z8tPKz93HOPr61ca8W/WW0CbfMUqrZo/sa08/zMvtzny88f/qqyYbe2WW4VQhVSxY9BxXLGNX7T0PUqVMK7ezjr+p5lfW76dM6Xg8racZbgGp/DU1pea9boLuFdrZwzgFvYZ6muP1fVbjV9Qkubh2OSdik/dFUa8xRipXPuvqsp0eWTtJo+hL8QR2slzdQLIIlyMjk+1chNe3F5L5srKv91VQYAqn4N8RR6lpNxp2sO5mjQIlxjcSh6Bu52kfrWj/Zd4gAWFpV/heIblYeuRXVWk6lnDY+Yw+H+qSlTrbp6N7W8v1Oi0a/+3WxjZQssXBCjAYeuKv7k8wRlwHPO2uftLVLGCVL+S3iluYyqRTy+WCvc5/GrFrfwwT2sepzQM0lqHW78zCSkNtIBOOcYP4120nPkV9zxMSqSrS5Nv1NK0u47y1FxHlELMvz4HKsVP8AKp65q+2eRa/ZcfZPKHl7eme/4561q6IJTp58zO3d8mfT/ChVb1HCwTwvLQVbm3NFZGT7pxSEknJ5NBGOtITitjjOd8W65NpkEdrZtsnmBZnHVF6ce5/pXCfaZ/O83zpPMznfvOfzrrfHVhI7W9/GpaNU8qTH8POQf1NcbXi4qUvatM+3ymnSWFTitXuegeEdcm1OGS2vG3zwAMsh6svv7j1967C0nMgKPyR0PrXA+BLCVGuL91KxsnlJn+LnJP6Cu3tD/pC/Q16WGcnSTkfM5nCnDFSVPb9TQoooroPOCiiigAooooAKKKKACoLtttufc4qeqt8f3ar75oApVK08Asmiuh+76E59f61EOvNPsIRdEz3D7njkZREowqEHHTufc/higBLKG6u7ON5b19nIXy1ALAHgknPUemKpvp8RVRAWt3XPzxsRuyMHdjlug5zn3rbnuVh+UDLenpWNfytFGJzM0MKn97sUFuSACMg9M9KAIDBbrYGaOCKDeoEjKP8AVgHBI9MHnPA71dDLIu+Ng6k9VORUcIhjd0gkBcHdJh8nJHU/pRJCJJPMWR4JOheLaCRnPOQQaAJk2ySFFZSy9V3cj8KZqO630a+kzgrbuQR2wpqnqtpH/Z9nBbxKzmZVjVuA2ASwY9cEA561abQtNYEfZFVSMEIxUN9QDz6fTigCYxiK3hiUYCIAAO2BiminytufnimYz05NAC0Ug60tABRSO6xJukYIvTJOKjS5ilk2RuWO0sDtO1gCASDjB6jpQBLSGkdlRS8rrGo6s7AClbIODQAlN6GnZ4pMc0AJ1oC4NOooAimwjRTsm8QsSeASoIwWGfT+WamJJ6nNIMhgRVe12o89usm4xPgJnPlqQNo+lAFilGKbnHWlBoAu2H3pPoP61BZf6mT/AK+Jv/RjVPYfek+g/rUFl/qZP+vib/0Y1AFiiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBunf66+/6+B/6LSrUH/HvH/uD+VVdO/119/18D/0WlWoP+PeP/cH8qAH0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRQTgUAFIc7TtOD71RlvHL/uuFH60JdyliNu7PYUAQzSPJJ85zjjiq1znyRtzxIjHAzgBwT/Kp3JLnI2n0puGPCHaxGFOOhoAjNxBuAE0fzYKkNwc9OelAm83P2ZBMq9XDgKT6D19/wDGm26RnT40EY8uRAWRvmyT1z685qbPT0HA9qAGKFuYSpUkMCrowwR2IIqGDbbpHHBIJ7YsUDqQWV/9ojg988ZqZ4YpG3SRqxxgk9/r6/jVe9igit3uAkUcsWGRiMDcMBQfY8D/APVQBbAycColuFkUNDFNIvXcE2j8NxGfwzTVa7cGVXtdpUeWikurd87uOvToafAhigWNiCVyBgk4GeBz6DA/CgCOKVReSQgOgcB1V0Iy5LbsHp2Bx9akuWZbWTy/vsNicZwzfKOvbJp7oJV2tnqCCOoI6EU0ac8skZM1xw6uMt8u4Hr079MZxQAQwpbwJDEMKi4+vvU0cZkbC1YFmxzk4544pFtJlbK4BHfNAF5FCKFXoKWiigAooooAKKKKAAnAyaqTXcTxyRxsGYDBGelSXMyJGVJ5PYVh21ibe6MxlDDnAHU59altpqyNIRg03J27Fo9Ka7rDC0kn3UGTilJxSOizQtHJ91xg4pu9tCY25lzbFJLyPUSIVjCup3qshyj4B4b88/UZ9qti5DzeSwkWQLuKlTgDOOvQ+1Q2enRWk3mCQu2MDIwFp9oVuf8ATFdsSAoqg4XaGOM+pzk59/Sop8/L7+5rX9lz/utieobwRnT7lZ22xtEysfTIxU2V37N67/7u7n8qq6pbtc6ZLEhw3BA9T6U5P3XYmkl7SKlpqeBzQvbzvFKMOhwRTK77UNAi1GXbLC6TjgMgw34+tXtM+Gumm6WHULu4ecxeaYkAUAE4GTz3ryadOVR2SP0SpmmHowUqj1Oe8GWzKLi6fhGARc9+5/pXpHhidpbKdQcxpNhCfccj/PrWLZ+GLaUwJaaighlQvGoiIYqDg8ex4rbs7uKyjt7Kxgxm+e2y56hBl3OO/BxXTh6VSM7yVj5nNMdQxEXyO7f4FDVPEsy3LQWARY42xuK53H2HQCruja1/arG2vY0MyDeh25DY9j0NYOq6TcWN4/7tmiZiUcDII/xq3oVs9nI2oXClUQFUU8Fif6VnCrV9r7xdXDYV4W8Er20fW51hKrGd4XYoJIxx6159qWuXmoXDN5zxRZ+SNGwFH4dTXa2t6L8sgwgx86sucrjsRj+tcRfaPc2smYkae3f5opoxuV1PTpWmKlKUU4bGOVQp06ko1fi6X/Q1fDOt3BvksrqRpYpchC5yUP19K641yPhnRLgXyXt1G0UUWSgcYLn6elddW2F5/Z+8cmaey+sfuu2tu43aHUqyhlYYYMMgisP7D4V+2Y22nm56eadufpnFL4uu5bbSEjhJXz32uR/dx0rhayxFdRny8t/U68uwM6tJ1Odxv2/U9XChFCooVVGFVRgAe1WbJSZS3YCuZ8I3ctzpDxzEt5D7UJ/u46V1touIQfWuynNTipI8bEUnRqypvWxPRRRVmAUUUUAFZWv60dEtYJltjcebLsKhsEDaWJ6c8DpWrWRrsEk91pOyJpFS9DPhchV2MMn25oAn/tZG1Gxt4VEkd5C8qyhugGMce+6nxa1pszyrFfQOYVLSYkHygdT9BXMnTL6HWjp8MUvkRW9yLafadqrIBtXPqCCPpinaNbySXWmJcLdJJaKUMTWWxUBUggv0I/nQBfbxdaz6BcX1i8LzwqWMBkBIAfbk47f41bOo2l/cOlpdQzNEPnWNwStYBif/AIQ+/wBL+wzi8j35/cHDgy5GGxzwf0qzqVhNLqkws4/LV9MmhR1XaoYsMLnpmgDQt720u5HS2uoZWj5cI4O0ev096qy6wLG2vb2Ca3ll8pGWFWz8m7aGJB5yCOeKo2EMs01p5gulktoHj8o2flqgKY2lu/IGMZ5qM6dJF4O+zwWLrcPbIZTs+dmyvy+v+G2gDcvNQs7ScpdXcMUh5CPIoYj8TUE93BEyvfzwWyJh1jklXJzkAt29cAZ5Gc8VU1Vrl7y9tWW48tosQiG2D+dkHIZyDjB4xxTtNs5F1CB7i3bjTYI9zp0bJyuT39RQBZnubW1W0leWGK3JJWXcqptK9BzznIPHpTbbVbeayku5pYYYFneJZDJhWAOAcnuaybeC5s4NHmlgmCxWzR4EBkMLlwclOvKgipVt2Wz86WO7tnW9mlhaO2LmPJ6suDwQeOO3WgDcVrTULNQ3l3MDHcrI2RkHqCOhqtd2n2KzkurO7uEFuDKIpJS0ZxyQc888jr36UaXcTTQxx3ieXOxdk/cmPzEB+9jsTnpTr/zL/dp1jhQrDz53XKoRhgByMt0PpigC7J82xtpBYZKntVVyLnZHGGaMPmRwcKQAeAep5x06VLJbhuJJ5pP73z4B/Lt7U8AAAAAAdABgCgCDdLbxkPE8yx5IdHyxUc8g8k4+ualSRZESSM7kdQyn1BFNuUaSzmjVdxdCpHseD+lLG6SxgwqQuSqgrjGDjGPwoAqiL7PfJLcTyOjyt5WZTtVmHClTxxyAR9MVYmWdriKSERkKrK3mMQcHb0wD6Uy4I225yGDSqwC4JYe35j8KsUAUbqKZkkaaRTK0bJbQwoSFYjG4nqSMgZ4A/GroUKqoqqqooUBRgACqt5cC3vbJs7N7NGzlgEAOCQQe5xxirJ/PPII70AOxQeaQdaWgAAxSYpc0UANquJTFqHkKu9ZVMhwDmM85OcY2nHrkE+9WSKgk41G2wMM0cgJJzleOMeucHPt70ATHpTR1p+KXFAFrT+sn4f1qGy/1Mn/XxN/6MaptP6yfh/WobL/Uyf8AXxN/6MagCxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA3Tv8AXX3/AF8D/wBFpVqD/j3j/wBwfyqrp3+uvv8Ar4H/AKLSrUH/AB7x/wC4P5UAPooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigCORo4Vy3A7Cqct3uXEYI96fcwyuzMSCijIqqsbv91SaAG0dOTwByT6USlYMmZ1jA6lmAH51C7C5iaOH5lYYMwPyr9PU/pQA20Ynzlw5USMysVwhBOflOP8+tWKVjuYk0lABQDg5FKKeImZcqpI+lAFUH7NJsIVYG5QqMBDwNuO3c/nV23iLSjcmV75qxBa4+aTv/AA1a6dKAECgAAAcUtFFABRRRQAUUUUAFFFNkVnQhW2n1oAZJcxxttYnOO1Up7lpTheFqtfSixUs43nOB70y2uFuofMVdpBwRU8y5uXqaeznyc9tCWjNKOtY/9rXH2vGBs3Y24qZ1FDcujh51r8vQ1yM0ClPWkHJwK0MBVOGB96jsRst44ccwDyiAOmOh/EYP409ika5eRQM4/GmS20UzZkD524OyRlyO2cEZouOzWpj/ANn3n277jZ358zt165rZubaO7Ty5C4Xergo2DlSCP1FRxPcRXAtpl8yMqWim384GOGHqPXvUGsTSQ6VI0JILMFLDsKxUVSi2dcqk8VOMHp0LbpDcM6iTDlSpETLuGe/QnIoSCJL4XY3mXyRDy3G0Nn061xUcjxSK8bFXU5BHau4QlokZxhmUFh6HFTRre0voaYzCfV7O97mZcaWkNnZwWlpNP9m3bJEuhFImTnrxkH+lJZ6IRpfk3heKb7S9yrQy5aFieMMevHX1zUPiXVpdOt44bU7ZZskv3VR6e9cnFqd7DMJY7qYPnOS5OfrnrUVcVGnLlsbYbK6mIpe0vbsdtc2b2unXRt5riaSUqXMsm44AwccDHFc8rlOnfqPWuhtNYil0eK+uD5ZOVZR3YelZep6zZXNtnymjZWyGxxjuTWdfklaSl0N8CqtNuk4ddwsNUt9OkaW+IihK4L4Jx+VbGj3Ony6fHFpN0txDCoXh8sPqOP5VwuvX9qNJljEqO8gAVVYHv1rmNM1K40nUIru0cq8Z5GeGHcH2NZU8S6fuvY9OplCxUHUTtL8Ge39etZE3irQYJ/Jl1a1Dg4OH3AfiOK434jeJ5HjttMsHaOGeBZ5iDgsG+6v0xyfqK87rerieWVoo58BkPt6Sq1pNX2SPfr20tdd0rYkySROd0U0TBgD6jHWuX/4Q3UfO2+ZBsz9/een0xXG+CfEM+i67DEZCbO5kEc0ZPAzwG+o/lXtaxuWwEJ/ChRp4lczWphifrOUT9lCV4vVaFfRNHjsLNbeIllB3O5GN7VtqoVQq8AUoGBgdKK7IpRVkeDOcpycpO7YUUUUyAooooAKKKKACmu6opLHHGacSB14qldTRyrhOSpxmgCGaYyyZPA7D0qOijGeBQBUjlEVxdG4lWJPNARGIA5A5Hc5/mDUxurfYj/aIgrjcpLjkev0qGW72I01payXTqnyuiYBHpuOCf+A5qSzSIK80DI7TkGRowMZx93j09+eaAK908MrwzpcSBoDuQISEk5BK5xg5AxjP4VcyHAZDlWGQR3BppuBIGCwTTryDhRtP4sQD3/L6VJA0TpkFVRByOmz2I7UAKiOegpSpz0OazZrjVNUiP9kqtpEkuwySOA5KsQ3y4PHHsf62ZH1xZURI7KRS2Wl+YADHTbnOc9wT9OOQBNWsZb3TUjiCfK251YAFl7gEg4Pvj8qmsntX0uKXToVggl+YIqBP0FRPp0k6n7dqNwzHgrAfLQDHIwOfzJPpirLFVRY4htRBhR6AUAJRSA0tADJo2lhZEkMbHHzDPr04556cetQraybSjz7YSOYokCj3GeuP161ZooAjjt4IG3QQRRHGMxoF4/CpM0U1vagBwOGGeRmqlkY0je2Q4MDuoUnnaG49+Mgf/WxVgGmyxtIySIwEsWdrMxAKnG4HHbA/SgB9BOait5vPtYpiuzzV3heeAfqBUtABSMxVGbazbQTtUZJ9h70p6UwSFMsf4QT+VAEdtPNeWwuITDHG5OwMPMOM9ThgM+3apYofLeSR2DySHlguMDHQcnjv+NNt/wB3bxq3UjdgdBk5x+GalBzQAtFFFAFuw+9J9B/WoLL/AFMn/XxN/wCjGqew+9J9B/WoLL/Uyf8AXxN/6MagCxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA3Tv8AXX3/AF8D/wBFpVqD/j3j/wBwfyqrp3+uvv8Ar4H/AKLSrUH/AB7x/wC4P5UAPooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqrclnJjWMlejehHpVqigDIt7D7OqPeXJlkUYDHIwMY6ZPPqff8ACllKl/kzj3NX5YWl3DhQe/UmomseBtJz3zQBSpQpJAHetCKzjRfnG4+9TLGqjCjAoApw2RbJlyo7Y71dRQihVGAKWigAooooAKKKKACiiigAooooAKDzRRQBnXdksn7soZFPPqc06105YVC7Qqeme9X6KVle5XNK3LfQiNtHtIVQCR1xWFqkEFkhuIo1aXcBnHQ+tdFUE1nDPGyOgIbr70px5lYulU5JpvbqctDrE+MSKkh9SMfyrSubgWtr5+0ksAFU9iRV9NJtoIyI4lz1GRn+dZCaZO104uzmNs5+b73pWKVSKtvc7HLD1Jc1uVL8ShHeARlXiVvm3HJNXxeNPM1vvEYcYTA6H0rHu0jEh8jd5Y4+Y8n3q7pqw29uLq7DffxGPWueEpX5T0K1Onyc9tenqaYiMLwNPIpVI3BZjjk7cfyNSZinhYDZLGwwwzkGuP1O/kvbx2ZiEBwq+gq/4bkkW8aNs+W6nGfWrjiU58iWhzVMvlCk6zl725rQ6NYwzCVY2Yg5Cs2QKmudQgtpAkpJY8kAdKW6vLeyUG5kCZ6DuazpIIdXmM1lOp4+ZW6itZNQXLTtc5qcZVnz4hvl7kfiLTH1WzhubL948YPyjqyn09xXKRaZezTCKO0mL5xgoRj656V6JbwiCBIlOdo61VfVlS4aPyyQuRnPU1jWoQk1KTtc7cJjq1OLpU48yWxiahpclhpFrGDvEefMI6bjz/8AWrIKl1YAbuDxXSrq07MRKqOjcbCvSs6Uosm07V3chelc9SMW7xPQw9SrFONRa739TlLTRoo8vcqHdiSF7KK0ks43i8v7OpiPGNnFSzBfOby/u5rRSWNohtI6fdrmjHU9CpWm9Tl/EmgXGpXC3dqUBjgSPyuedowMH6YrjYLea5uFggjaSVjgIo5r1pMKCWwPrWXFpUVt4jfUYNipJHtZFH8R6sPyH61q1fVm+HxjpwcGtlocnpPhvUZPEtjZz2zxmSQOT1AQHLHI44H9K+g412xqD1xXgni34iXGiapDb6EzJcwPmdpE+Vh/d56+ufYV6Z8PvHlv4z0wllEN7DgTRZ/Ue1ehhoqMfU8POo4mtGFaUfdX6nY0UUV1HzAUUUUAFFFFABRRTJlZoiIzhqAGXNv9pjMbFTGwwysuQay5bJoJVNusULhSuCnyuO3Qjp6+5qZ4pI+XUj3qxuMqL54wOz5oAzEmmnjGyLyucGUkMpAPJUZyfbIHrzTzFI4xNNuXusabNw9+SfyxTGuYrCObzEeRhPtjSMZ3F+Vx2HXvgZzTF1lB5kUtt5d6r7I7cNkyHAOQfTnk9sUAW/QDgDoB2qJ4EkkLt5gLddsrrn8iKz08T221kmVLi48xo4xbZYSEKD35HJx9a04p0u7bzkQxlXaN0OOCDg9OO1ADlUKoVQAoGAB2qKa0t7oMLiFXLDbux8w+h9qlqG7H+jFssArKzFWKkKGG7kc9M0AUdODaXqbWhJaK5c4VnxsYBjuAxjDADoeueK12LIxUE4qtLYQSW9wPmRyoIlLlmUrkqQTnoTUkErXOnWtxIMPLErsPcgGgBcYpBzTqKAExS0UUAFFGabvoAdVe9v7fT7fzrtyqlgqqqlmdj0AA6mpi3FZN+wTxNo7z4EW2ZUJ6CUgY/HHSgCzaakl3ceSbW9tn2lgLmAoGA9DyPwqe++XTLksjMGjKAAZ3E8Ae/JqhPqGo2mvPaOIJIXgmmjVY2DDZgqCc4PWs3T7e7uLKw1C1trprx2SWa4ecFZVJ+YEbumM4GOMUAdEs5lYF7eeAscASJwOPUZA9OtTBG/un8qwEie61q+W5uZhbJfqqxpKV3NsBGfYY6d8mqV5bpNpWuXkt1OLm3urhrcC4Zdm0g5Cg+9AHWFTjODj1xTRG+4FR+PpWFHCltr+lSWc80styJkuBLMxDkRhhnPTk9hVPUm8zTrjV9OhulZJC63jXhGSHwcJn7ueMUAdBC2ySWJIJmhjkIV448qB3A6E4bd0BwMVYUhl3xncv94VkfYIb/wATahbTvMLaOGDbDHKyLk7vQ1Rgu3h8N6ffNdO8ljM/moZMs8G9kOR3wMc+1AHUAnHI49aWszSna+vL++VyYC4ggXPGxPvNj3Yn8q06ALdh96T6D+tQWX+pk/6+Jv8A0Y1T2H3pPoP61BZf6mT/AK+Jv/RjUAWKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAG6d/rr7/AK+B/wCi0q1B/wAe8f8AuD+VVdO/119/18D/ANFpVqD/AI94/wDcH8qAH0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFULyR1mwGIGOxoAv5HrSMFYYOCKyCSepz9aAxU5U4NAFxNKtklZxGMt14qjrNjHOqBJFjK9Nx4qwLqYkDf8ApXD6vezXWoS+YzbVYhVz0rmrzjThqtz0cFSqV6t1K1vma8nhsSKGFwA564Hy1Yt9IeCaM+apRMciqHhq7lad7Z2LJt3DP8JroamlCnOPOka4qviKM3SlK5yXiDzP7Xk8zOMDZ9KPD/mf2tH5ecYO/wCldPc2dveqFuYw2Oh7iltbK3skK20YXPU9zUfVpe1576Gn9ow+rey5dbW8ifHcdqxJfEHhqC9/e6laiYHnD7gD9RxXLfEzxBPDJHoto5jRoxJcFTgtnov04yfXIrzinWxCUuVK9jvy7JXWo+1qTa5tku3me56r9lHh+61DTRFKVhaSOSJtysR9OK8C1rxFf69dRT3rqrQjEaxDaF9x3zXV+Dtck0vV0tZWLWN43kzxE8Ybjd9Rn8qxPGnhK58Mau6ld1pKxaCQenp+FTzKceZKx62Aw6wdd0akrt6xb7dV8jV8OeMllVLPVyfNzhLjjBHo3v712Uc0bbArqSxwuD94+1eJ/SvUPhf4UupruPXNRLC3hz9nRmzubpnHtWSo88tDbMI0sPSdZu3l3ZU+Kkl5pulafDICnnTeYV9QvQH8f5VJp/xA0aTTEe6laGZUAaMryT7V1XxT8OP4g8MBrbBu7RjJGmeXHcCvnp0aNykilWU4IIwRXTKjG3KcmXShjMOnLdNml4h1Ua1rk96q7Ec4UH0FdN8IL2a08f26RMQs8bI49eK4avXPgt4SupL6TX7iFliRDHb7hjcT1P0rWKtojqzGdOlg5qW1rI9vtJHlLF2Jx2qzUNtAYVO45LelTVsfnYUUUUAFHTrVZ76Jbz7NyZPL8wehGcdfy/MVWlneXqcL6CgCy92DIFj6f3jWI+tta6hf28++abzo/IhBHzKyqOPxyT6VdqO3l23d1ghXLKxQ9fu43fjjHHp9aAH2+ryhoor+zltZJSFUMRIhOM43D8euKW51WG3mKNKZbggbbeIZY56cdvqaJkhuovKvIUmjzna65GaW2ht7SMQ2UCQqTyFXFAGX/ZRu5pRqc86SzBZnWGYhV54THTjaORzwelbDTkN8o4qlbu13Kl2FUK0WBh9zMCQRkYwMYP51ZwTxigBFW2Eiv9miDqSVYKMrnqQfxqkzz6Td7QnnWFxKWVY1Z5IsqS3AByu4f+Pe1XHKxIXlIRfU1CrtcMjoAkcT8MSd+R1G3HGff8qAGNqao6eZYTx27MEaaUBdrE4HB5I6c+9SrFPPAHZvKjdchEX5wD2JOf0H41M5juIWiuo1ljbhlYZBrMu8aXqNkILyZVct5kcsm5EhUZJ56Y4AOe/egC7du9vBHaWYAuJxsjBG4IowCx9gD+JwKkhhFpY29oG3+RGqbsYzgYrNvY5dbuopNKuGt44MqbxW4cHBKqB94ZUZOQPrzWvKwZuPzoAjoFFFACE00nFLSGgBM0CiigAx82KpyJb6jut9QgaVZNwWN0YIhUk9f7+Ocj0P42pHWOJ5ZMhI1LMQM8CmMJJp4RJDJHHETJl2UktjAAwTxgnP0FAEVnpNlYXHn20JE23aJHkZyB6DJOKami6dHMJI7cr828IJXCBuuducVdyRTloArm1EVw1xaCJJZGLShwTuOAMg5+U4HofpVcaZC9lJHNbxPLMztMqOwWTeRu6ngkAfQ+1XjQBQBGiQ3Tw3AhYPaMyRh2+aIkbSCAcdKqHQdNbzA1sWSQkmMyvsBPUhc4H4VJPG7axa7ViVSrOzjKyNt/hz3ByOParinsaAGpBEl3JdJGFmlCh2yeQvQfrVS5tZINHey0q1iZZA6FZJSFQNnLHOSeT0q/RQBnLY2+maRBHFhXtECxSdC7f3cZGdx7e9aILEDeuxsAsoOdp7jPemTW8V1EYbhFeNiMhvbkVBFPcRW8Ru7WUKqBZZi6kg45bAJ4/HPtQBq2H3pPoP61BZf6mT/r4m/wDRjVLpskcvmNFIki/L8yMCO/pUVl/qZP8Ar4m/9GNQBYooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAbp3+uvv8Ar4H/AKLSrUH/AB7x/wC4P5VV07/XX3/XwP8A0WlWoP8Aj3j/ANwfyoAfRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRTZJFiQsx/+vQBRnuZBOwU8A4xiq7MXYljkmlkbe7Njqc0g+8KAILq8trKPfdzLEp6ZPJqK01Wxvn2Wtyrv/dPBrhfEN3Jda1P5hOEbaq+gFZ8UrwSrJExV1OQRXmyxrU7JaH0tLJYzoqTl7zXyPVehrL1DQYr2czRSeU7feBHBrQtJGubOCUg73QEgCrS2sxGdpFd0oRqR948GnVqYebcHZmbpumQ6ajbW3yN95jVqSRIozJM6xoOrMaleJ4z864rA8TeZut+vlbT9N1TNqjT91GtKMsXXSm9zYt7mC6ybaZZNvUDqPwqZsgdK4yxdo5i8bsrgcFTitawvLlrxFLtIGOCCc1lTxPNa6Oqvlzhdwlou5j+JfDlvd+JjqF2vmRyQIFQnA3DIOfwx+dcjr/hxFurcaVCd02QYweBjvz0r2S406OeDbdAbc5Tnn8K4ptMjj1KW5innlLDYqylcKue2AKwxFO0ubuerlmYScFFv4Vby8jidM8Lai3iKxtpYCFklDM6nIVQcsc9uK6Lxy7TeJZY5hujjRFQEcYxn+ZNdroelXKXTSyp5Y2lQCetM8U+EG1iNLi2dVuo128jhx6H/ABpOhJ0Xbe5bzSEsbF1GrJNX83/Vjy2z0mG4uQsUMYbqW29K7jw/c/2JYXMDsZFP7yMY+62Of6VS0zwnrVteEyWqhMYLeYp/kc1u2Xh66mn2zpsTv71jSp1Yyuk7m+OxWHqxcJTTj5M5S4uprqZpZ5GZmOeT0qK28IaP4pvmi1KFlm25WaI7W49fWtm88LajbXBSKEypnhlro/Cvh2WwkN1d4EhGAo7Cqo06qq6r1MsTjKMMO3Slr0sYWl/Bfw9YzrPLFJdMpyFnf5fxAr0O2torS3SG3jWONBgKowBUtFetax8tWxFWu71JNhUUtwkXB5PoKlrMuW/0l8etMwJHvJG+7hRULOzfeYn6mmA0tAEF7lb7TZoyVkaQwk54KlSxB/74GPcValXa/wBajmt4ryFIpi6lHDo6NhlPqD9CR+NZlxf6jatFY+Q0kxbyknIP7zkEEHBAO3dnPpn2IBqlSOoP5VSvbyGyu7d5958xGiRY4y7M2Qx4HsKiur3UtIhhmv3huICRHKUjZfLP95jk8djx3H0qrrc8gvNKcN9hdZ5MPLtIB8s8ckA+mSe9AGpaX0F95n2dmLRELIjoUZCemQRTr65WwsZrqfcqxrnhcn8j1rmZbyVbfUSk63LGWAXF4jhB5ZJBXK5C49RnhqdeRBNH1Xy2hS38lWjghuDNsbJBPTgEY/EUAbj2tpaZLRyvMpMrSxId4yME5UAAYHTvjoTUrWsazR2832ucspILsSq455IxyffrWTqUttJr06ajMURY4zbIzsit13kYIywIx3+lO0yWaS50ANJIN1nKxJbO84Tn9SOaANcWFsTg2sZH91lyBzngHpzzxUGnxeUZY2gmimYCWTeQV5JwBgnp+Z61i2moZ0XQ4xcxvN9rTcgk/eHBbIYenSm3UKPZ312WnFzHqZVHErgIplUY646E0AdUkRLjIIHXkVni2TWrq9kmG6zeEW8T4AzzlmHrzjB9R+cNhC9nrV9b2Q8pWiifbNI0iqSXBbrznA4zRCmtaRH5UcVvd2ykiMBmDgduxwPbtnrQBZsjcWOo/wBmzkTQSRmSGXaqkYxuUgY9c9PXk9rhGCRWXYmWz1u4l1W32yXRQRTr86oDkeXu2jHIHHuM1qOGR+enY+tACUUUZ9aAG0mMdaV2VVLMQqqCWJ7AVEJmlUG2ifkZEsy7VH/AfvH+XvQA3zh9sMGzAB2iQtjLYU4x9GH5HipiMHB4+tRw20UMIiZUlDKBKzJkyn1bOc8+tV1Zc21jJk7fklX++qpkEccjpke+PqAPlaO7jNvDIsivxMUOdsfcexOMfiTVnqOBigsST9aWgBMUvSig9KAG55oZnWNmjj8x1UlY92N59M9qUVHc5MIjR9jSOqAgkHGct0x/CDzQBXsohLDb3UzytKhbKGYum4Flzz3xnkYH6VdBpiRpBEkcKKsSjChegFKfagCTNBOKjDetOzQAuaMnPBxSUnJ9qALun4HmYAHToMetQ2X+pk/6+Jv/AEY1Taf1k/D+tQ2X+pk/6+Jv/RjUAWKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAG6d/rr7/r4H/otKtQf8e8f+4P5VV07/XX3/XwP/RaVag/494/9wfyoAfRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFNlYrGxHXFAEVxeQWw/fSKv8AvECqU05n2sMFexBzmvPtTvJb6/llmYn5iFUn7o9K0/C13KL82mS0Uik7f7pHcVxRxalU5bHs1crlToe05tVrY6mlxgjkAnoCetTLaydSuAPXvXI3/nf2hL527fvOPp2xW9Wp7NXscOFw/wBYk1e1h2v+GJLy6a7sNu9vvxscZPqKzdP8KzteINRZII85K7slq7WDzPssPnf6zYN2fWq1xpc1zeebG3ysec9qwqYeDfOkehQzGtGLoykkl1N62gjghVYhwAAKmpkKGOFVPJAp9dp4r3EZVdcMMiq0unW80JjlXepPRuatUUbgm07oz4dGs7dGWGJVDdflzn86nisLeF90cag+wqzRSUUtkU5ye7GuiuuGFZn9g2/2rze2c7c8flWrRQ4p7hGcofC7CKoVQB0FLRRTICjGOlFFAAQD1FFFFABRRRQAVmXK/wCkP9a06zJmzM596AIxRR3ooAKkEzAY4P1qOigCLUoPt2jXkBTzGaIlV7lsZH6gVWbWNJnhgF+8RkYAOrx7vKY9d2R8vPHOKvqxU5FP3LKdskasCecigCNUiEA+z+WYGHGwDafy4qNba3WFokt4kjY5ZUjCg/kKoRajBpupXlj9nncvKZEhgjL7VKplsdgWY+2c9KuxanZSyxoBMgkO1HlhdFZuflBYDng0AV761vJSFtLqVI2RsoFRiD7M3IJz7/hTLZy8UVrp0X2cW0SqXuU/eRggjAUjk/LycitSVPLyxIC+pPSqcUwn1B5YpVkhFui4U5+bc3P5Y+vFADU05QxcPGZydwZYERS+eGIAyfz7mpoZY7i3LCJVBbMkbIBtfgnIx196kJzUM0R8xZ7ZI/PyN/GPNXGNpI/Aj6UATZG4ttG4jBbAyfxp3mMBgE1Qt5by6xNG1t5TSMqpksMD5chgOcsCeR0qUXKeVI0qvCYnCOGw3JxjG3Oc7hQBPdBZ9LuUng+0p5ZzFnBfjpnt9ah0qRLjSwsd+l7IvzMwcNszyF9fbnmpraUSEmBg+CVbHYjqCOoP1rPv76O08TWxij82QwskqQtl8HDAle+MH8/pkA0Rk9KiaRmlaKIDKqGaRvurnp9TjPT2qvcSzTRTwtYzbZYvkIZTyezDsc+5qeOJIhlIREzgFwOucdPwyenFACfZtyN9puJZi4wyhtiY9AB2/HnvmpSxJzSrzSkCgBACzCoVk8+aJ4kby1DfvT911I7c+oHX0qZG2uPQGq9krQQfZJjueHlZM/6xSTzjtg8Y7cUATgetLTT1pymgAJxSA5pT3pBQAuKimtYrooJlZtp4Acr1x1x1HHepRyQKrp9omQsZkVGLfcTDqMkAA9M8df8A62ACK1uNkcheCZYTO587jaBuI+7nIHA7epPc1cwDypyOxBzSoQmAg2hRgAdhUIWS2aRlRp4CxfYhG+PPJAHcZyfX60APIozTDPulSOBBIzx+arMdq7fXOCT1HGKZJcvFu86zmwo3GWIhowAMk5JB/DH0oAlllS3UGZwgIJ5B6DqfYe9SYI4PWq4hiae/TaPnZFeQYDZ2Dj8Bg+nPT1cn2vGGa3c9PMYtk+5AGPw4oA07AY3+4H9agsv9TJ/18Tf+jGp+lCYLILh0dhgb0XbuHPOOxpll/qZP+vib/wBGNQBYooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAbp3+uvv+vgf+i0q1B/x7x/7g/lVXTv9dff9fA/9FpVqD/j3j/3B/KgB9FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFBIAyelABRTY5VlGUOadQBDNcrEdo+ZvT0qrLcvJkDgelXPJTzTJj5qZcQeZH8o+YdKAOV1Dwqb64a4snEZc5dGHGfUYrS0Dw2ulMZpm8yZhjOMAD0FbVvEYo8HqTk1LWKoU1LnS1OuWMrypeyctAqJraJ3DsgyO9S0VschDLbJIB/Dj0p8cYiQKOcU+igAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKRnCKWY4AoAGYIpLdAM1lylWclM4PPNNvmlnmWW3kZSnSNj8kg9D6fXtTIpVmUlMgqcMh6ofQ0APFFFFABRRRQAUUUUAP8/GTtG7pmoNSMU2iXn2tN8axMzD6DP55FONEryx6ddPbrvmWJiiYzkgHHHfmgCha2yw6Xp0kdybhQyDLHenzNz9MZIH4CtNFMmdxwo6k1jQ3lnaaR5XmThkk81mlhZFDh9zLnAUc54FWrmZdVtxa2EMk0LSx+ZPwIyoYFhyctwCOARQBPaX1lfzPDavIXRQx3RsAV7EEjBBpt00sawlJBEryqjNtyw3cArnjrjseCfSq9ir41aazCi5a42hXUBgqgKBj0wDtz6896ltYbeOCPyIseWAoLqN4K8c+9AEwQRrsUEAepyfxPc002NvcySSXCttIUv+8ZV+XoSAf5+lPFQ6nmSKHTYwxe6cCXaD8sWcsSR2P3ev8VAFG81uWKC5eyt5me7kEdnNgbWY/JnOcdRkZ6/StiGCPT4WSIszu292kcszHGMkn6Cq2uxr9iguCu5LO4Sd9uchV6kAdcAnirW5LqBLmBt8UiBlOOoPINAEe4kknvQOlGKO1AC5xSEk0gFOPSgCK4lFvb+Z8pOQAGOO4yeOeM54psFu0E80k87TyuxweVVFOPlC5IHTrSSxKbiOV4jOoGwLt3eWeTuA6dgPXpUn79/uIkQ9ZfmP/fIIx+dADutPAxTI5PMQ7k2OjbXXOcHrwe4wQadQAMVRS7sqoBksxwB+NQC5t2bCzoeduc8E/XoaWVM3do2A3zOpDAHHyk5+uVA/E08oiwmIIoiKlSgGBg9eBQBCVlkuLiNpCkSSoMKvJG1SQGzxznP1qcABQFACqMAAYAHpTY0VRhBgE5PJOT65NSdKAAUqnawPpQBxSHigCmbBzHHDJdfuYVCx+TGFYY77jk5x3GOtXGAcMHVWVgQVIyCPTFFFAEcUEUEflwpsXOSMk5OAOp+gp4GKWigC5p/WT8P61DZf6mT/AK+Jv/RjVLp3/LX8P61FZf6mT/r4m/8ARjUAWKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAG6d/rr7/r4H/otKfLdLZ6as7qz7VUBF6sTgADPckgUzTv9dff9fA/9FpT5bZbzTVgdmTcqkOvVSMEEZ7ggGgCjpuuyXsyJcWZhSU4ilQuVY4JxlkXOQCcruHHXpmW51S5tNQijmslFrLN5KSibLk7Sc7Mfd4PfPtUGmaA1jcrLPcrKE+4iIygcEDO524AJwBhRk8VN/Zt6dYa7ku7eSI/Ksb2zbo0xyqtvwCe525P4Ch+QEFnr9xeqRFp48ySFbi2Tzx+8jZsZY4+XHBIG7g8ZPFMPiG42+Umnq14sskbR+f8Au8RqGYh9vPUDoOeuMUQeHbq2hdbfUlWRYFtreQwf6qINnBwwyxHGQR06U99DumtbdUvLeGe33qjxWpCbHGGBUuSTnnOevY0PbQfUfqPiBLLQ7XU4YGnjuChC7tpCsM56HOB2px16Ma++n+UTGlqZ2uA2eQR8oUDnhgfx6VJLo0T6fY2aPtis3jIBXO5VGMfjWfB4VaC3KpqMnmm3lh84Jhvn2BT1/hVFHv7UdX/XQS6F4+I9MCgmWUMZDGIzbSCTcF3Y2bd3Q56VEviewN+YGfETRRSRzKrMrByQM4GFGQBknvWcnhq+sL62ksJ4cm4aWRxBhI8xbPu78nJA75561MvhN490ceoYgmgjhnVoMs4VmYkNn5SSx7HFAdDU/t3TvOmi88loQxbETkHb97acYYjuBnFTHUrX7LcXCSb47fO9lUnJAzgHv17d+KzU8PzxT7or8KkTTPbKYMmNpCcknd82MnAwOvOat2ejRWmly6d5jPatkRrjDIpHI3d+ckH3o6B1K1lr8txdbLiwaGEv5YlUswVicAElAp54+Vm5/OrF1qVyL97PTbNLmWGMSSmSby1UHO0A7TknB9B71Vs/Dz29751xeecitvCqjKSc5GcsQBnnCBQSAatXWm3Bv3vNOu0tpZoxHKJIfMVgM4IG4YIyfUe1HQCJPE+mGKJpZXiaSMSFGiY+WCxXLEAgAMCCScD8asNrenpcSQtOQ0e7cfLbblRlgGxgkDnAOax38NXP2uS0t5/KsJLFLeWV1DtId7lscjDHdnOCOelTDwlCt5czI9uFmMrhjaKZlZwQf3h5KgkkAAHtnFD2+8CxN4msv7PmubMmcxeXlXVogA5wrksOE6ndyMA00eIWiNot3aAfap/KSS3l86MjaW3BgvtjBA7noM0+LRJ7YFrS/wDKl+zQQBjCGH7vd1BPIO7oMEetQJ4aYTJObmFJ/tQuHEVvsjPyFCAu44JBznJ5ofkBbHiTS2t0nWeQxyAsrC3k+6MZb7vC8j5jx71K+t2CXDQeazSLn7sTFSQu7aGAwWxztzn2rKl8IRyW1jH51vI9rbfZi1xaLKCP7yqT8re/I55BqaHwrBBqrXSG22NJ5nNohlB24wJOy98AA++OKH5AN0vxXHqV5FE1q0EctoLgSO3QnPy4IHYE574qWw8TwXOkC/u4JbZS7rsSN5iAp+8dq8DHr0qovg8ixEDah8yxxRLIkOCFQOp/iPVXI/X2qW88Km5gMK3gWJpZnaNodykSH03AZXsTkc9KHu7AaA17TTeC2FxmQsq8RttBYAqC2MDIIxzzWjXNWugXovZ4551WzE1vIAIxumMcacg7vlG5eQQenFdLT0FqFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUjKGUg9DS0UARwwJApCDFSUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFVZ5Xgk4bcp7HtQBaoqhJfMwwg2n1zmrNq7yRZk554NAE1FFFABRRRQAdKrTXYTiPk+tQ3UrGUqCcDjFV6AJmupW/ix9KhdmY5Ykn3NGaOKAG1FO6xTQSuQu5/JJ9dwOB78gfTmpiMVVnVW1Kx3O6lTIwC9D8vf06HH4igC2eKKKKACiijNABRSZozQAuM0KSjZU4pM1FcSyKY4oAoeYlQ5bGzAySBg5OPwoAdaS7ftBX+Kd8qf4cYH64z+NTec7MvOBntUaxiNAqdB69Se5PuaKAKV81xbapcTw2szmSCNFliTdtUMxfjPJAORxzwMVctlgms1NkxZEJU7gQ2R1znnOfWpFmde+R71XurEv51zp880NyVL+WjgJI+ONwII7AcY96AJkwHw+QB7VXPmReJxJO2IpoBFBheCwJYgnPXAz05H0qa1nF95iTRmG5hIWVMg4yMjkexqlr0tq2imJ543mZlNuFYbmcMMY79eCR2zQBeE8yzSrBCsmw4YtJty2AcDg9iPxrO0t7yKy86yCy2e5lS1mba8CqSAN3PHHQ9PXitN7JBbtDCrIJFO9lb5iSMZ3Hqfc1nNd+ZDHYW0zHz5zHHMgAzEoUuQQAOhKgjv9KAL8UoutPt7oJ5ZmjVymc4yM4pCafKqW1ukahYYIVCgk4AAGBVYSzPuMNodgHDzSeWX57Lgn88UAS5xS4qNZ43m8n5o5uvlSDDY9vX8PQ1IPegBy8UtNpw6UAQs4hujvJCT7cHBOHHGPbIx+RqSoZJI4r5DOSAyfu2IJVCDhs9hww5+opYrjzrmWERSL5YPztjBIODx19wfSgBLnesayxDc8LiTZj74wQw+uGOPfFPeWMRJJk7JMbMKWLZGRgDrxUd5Mbeynl3bCq4DE42knGf17c+nNQ3EEd5GttbxxNbxxgZmRiARwoHTnAOf/r0ASreQ+YkZEySMQNjwsCMnAz7Z4zVnHODxUc0Mc9qLebLx4UHJ6kd6hh8u11D7MrsTcRiQB2ySwJySe5I7/7IoAt9BTTzTyCByKZQAUUUUAFFB6U3k0AXtP6yfh/WobL/AFMn/XxN/wCjGqbTv+Wn4f1qGy/1Mn/XxN/6MagCxRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQA3Tv9dff9fA/wDRaVZWEqoVZXAAwOB/hVGSytJpC8ttC7nqzRgk/jTP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8APlb/APfpf8KP7Nsf+fK3/wC/S/4UAaXlt/z2f8l/wo8tv+ez/kv+FZv9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAGl5bf89n/Jf8KPLb/ns/5L/hWb/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBpeW3/PZ/yX/Co3tFkbLyOT+H+FUf7Nsf+fK3/wC/S/4Uf2bY/wDPlb/9+l/woAt/YI/7z/mP8KlSDy12pK4Hpx/hWf8A2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hQBpeW3/PZ/yX/Cjy2/57P+S/4Vm/2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hQBcNkjMSXfJ5PI/wpPsEf99/zH+FVP7Nsf8Anyt/+/S/4Uf2bY/8+Vv/AN+l/wAKALf2CP8AvP8AmP8ACj7BH/ef8x/hVT+zbH/nyt/+/S/4Uf2bY/8APlb/APfpf8KALn2CP+8/5j/CmyabFL5e95f3b70w2MHBHb2J4qr/AGbY/wDPlb/9+l/wo/s2x/58rf8A79L/AIUAXPsEf99/zH+FH2GP++/5j/Cqf9m2P/Plb/8Afpf8KP7Nsf8Anyt/+/S/4UAXPsMf99/zH+FJ9gj/ALz/AJj/AAqp/Ztj/wA+Vv8A9+l/wo/s2x/58rf/AL9L/hQBb+wR/wB5/wAx/hR9gj/vP+Y/wqp/Ztj/AM+Vv/36X/Cj+zbH/nyt/wDv0v8AhQBb+wR/3n/Mf4UjabCzRszOTGxZeehwR/Imqv8AZtj/AM+Vv/36X/Cj+zbH/nyt/wDv0v8AhQBc+wx/33/Mf4UfYI/7z/mP8Kp/2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hQBb+wR/3n/Mf4Uq2SKcq7g/h/hVP+zbH/nyt/wDv0v8AhR/Ztj/z5W//AH6X/CgB93oNleyNJc+azMmxtshXcvPBAxnqaanhzTY1jEduqGMqysqgNleh3YyfxpP7Nsf+fK3/AO/S/wCFH9m2P/Plb/8Afpf8KALhsVMRjMs23GMh8H8+tVU0CwVYl8tnESlEEh37QSCRzn0FN/s2x/58rf8A79L/AIUf2bY/8+Vv/wB+l/woAmj0Wzim82OIK/Zgq/L9OOPwqb7BGerv+Y/wqn/Ztj/z5W//AH6X/Cj+zbH/AJ8rf/v0v+FAFp9OifZueT92wdcNjB/yTxS/YI/7z/mP8Kqf2bY/8+Vv/wB+l/wo/s2x/wCfK3/79L/hQBb+wRf3n/Mf4UCwjHRn/Mf4VU/s2x/58rf/AL9L/hR/Ztj/AM+Vv/36X/CgC02nQsQWZzgEdR0PUdKik0W1kVA3mZjJKsrkMCRg/MOenHX0qL+zbH/nyt/+/S/4Uf2bY/8APlb/APfpf8KALEmlQSwmKV5nQ5yDIec08afGABvk4GBkiqn9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAFv7BH/AHn/ADH+FIdOhYqSXypyDx/hVX+zbH/nyt/+/S/4Uf2bY/8APlb/APfpf8KALEGlW9usghLqJZDK/wA2csep5p/2CP8AvP8AmP8ACqn9m2P/AD5W/wD36X/Cj+zbH/nyt/8Av0v+FAFv7BH/AHn/ADH+FH2CL+8/5j/Cqn9m2P8Az5W//fpf8KP7Nsf+fK3/AO/S/wCFAFv7BF/ef8x/hR/Z8X95/wAx/hVT+zbH/nyt/wDv0v8AhR/Ztj/z5W//AH6X/CgDQht1g3bCx3dc1Rsv9TJ/18Tf+jGpv9m2P/Plb/8Afpf8KnjjSGMJEiog6KowB+FAD6KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAP/2Q==)

1. Accident hotspots in Great Britain

Analysis of every district and highway is a humongous task. For the ease of analysis, the research was focused on five districts and highways that are more prone to accidents. The top five district local authorities in terms of accidents were: Birmingham, Leeds, Westminster, Lambeth and Cornwall. The top five highways in terms of accidents were: Kent, Surrey, Essex, Hampshire and Birmingham.



1. Top 5 accident-prone district local authorities

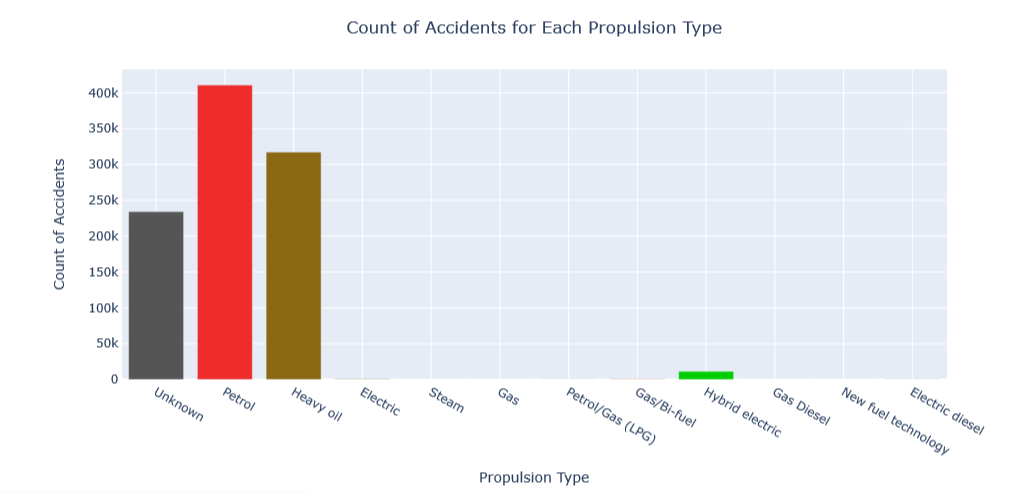


1. Top 5 accident-prone highway local authorities

Both the district and the highway local authorities for Birmingham were present in the top five accident-prone districts and highways. Thus, more awareness regarding road safety should be made available in Birmingham.

### 4.2.11. Propulsion Type

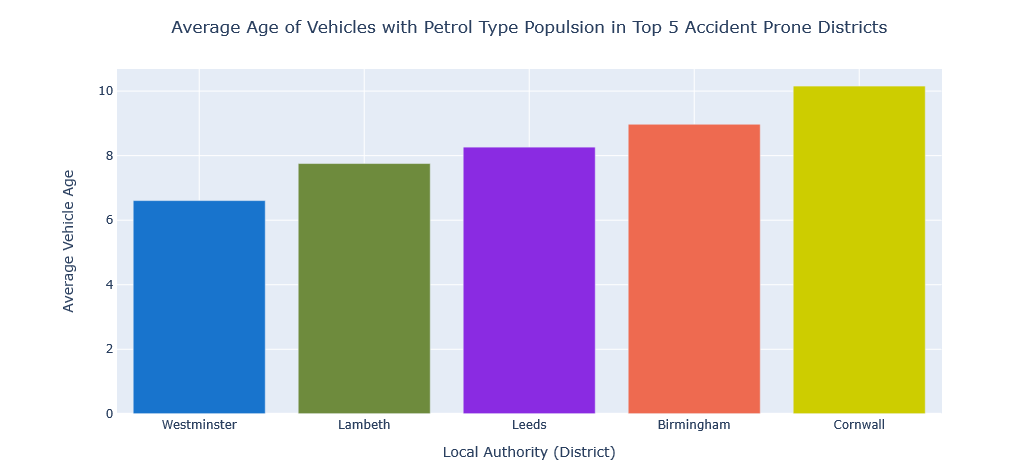
To understand the role of vehicles in road accidents, the count of accidents for different propulsion types in the vehicles were plotted on a graph.



1. Count of accidents for each propulsion type

From Figure 4.2.11.1, it can be seen that the vehicles with petrol propulsion type were involved in the maximum number of accidents. This could be because petrol is the most commonly used propulsion type for vehicles and therefore the majority of the reported vehicles had petrol propulsion.

Along with the propulsion type, the average age of the vehicle is also vital for the investigation of vehicles. So, the average age of vehicles with petrol type propulsion for the top five districts was visualized that can be seen in Figure 4.2.11.2.

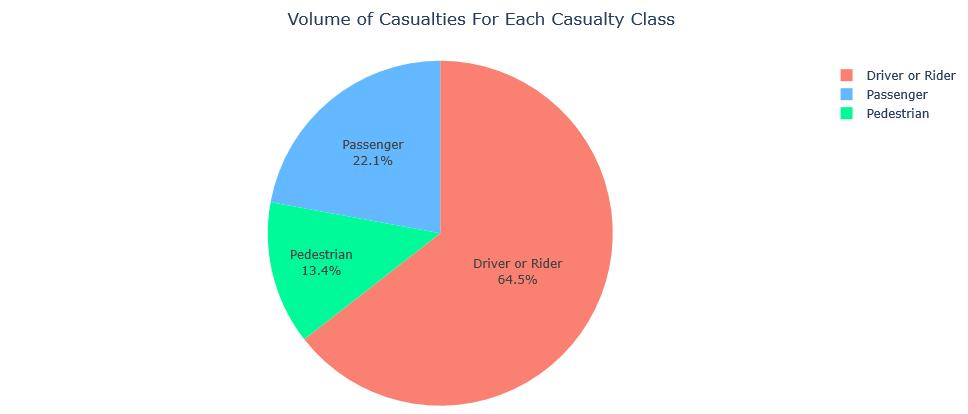


1. Average age of vehicles with petrol propulsion in top 5 districts

From the above figure, it can be seen that the average age of vehicles with petrol propulsion was highest in Cornwall but among the top five accident-prone districts it was at the lowest spot. Hence, it cannot be concluded with certainty that aged vehicles result in more accidents.

### 4.2.12. Casualty Class

To understand the proportion of casualties among different classes such as driver, passenger and pedestrian, a pie chart was created which can be seen in Figure 4.2.12.1.

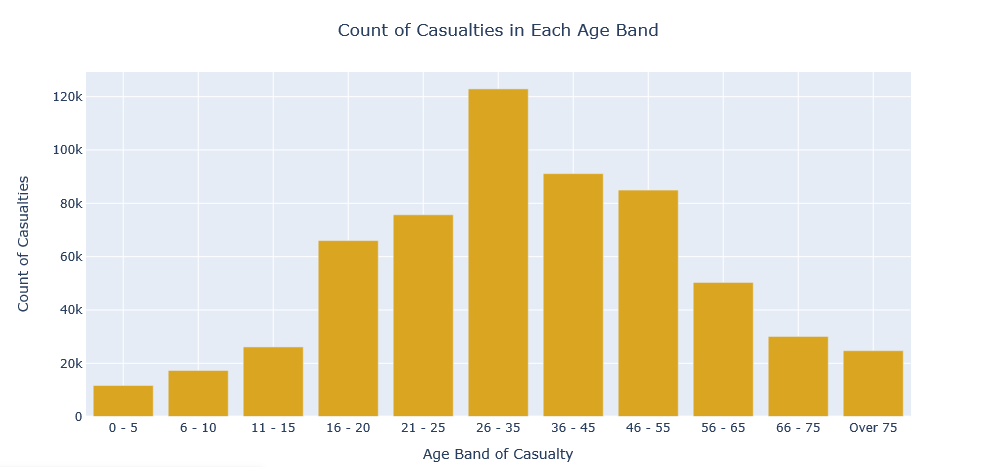


1. Distribution of casualties among different casualty classes

The majority of the casualties belong to the driver or rider category. Therefore, additional safety measures need to be put in place to reduce the casualties in this class. Apart from the introduction of additional safety measures, the implementation of existing measures should also be audited frequently to ensure they are being followed.

### 4.2.13. Age Band of Casualties

To spread awareness about road safety measures among the public, it is vital to identify the target age group. To identify the most at-risk age group, a bar graph of the number of casualties in each age band was made which can be seen in Figure 4.2.13.1.

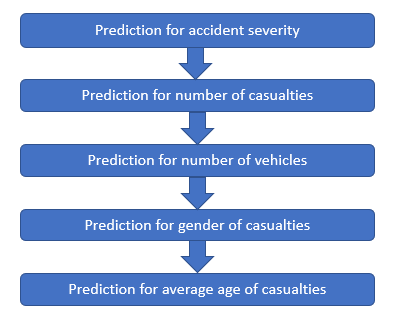


1. Count of casualties in each age band

The people aged between 26 to 35 years are most vulnerable to death in a road accident. Thus, awareness regarding road accidents and the different safety measures should be primarily targeted towards people from this age group.

## 4.3. Predictive Modelling and Analysis

Based on the nature of prediction and the input variable requirements for the prediction, the predictions on the dataset was made in the below sequence. The reason for this order was that in some cases, the output of the previous predictive task would be used as an input for the next predictive task.

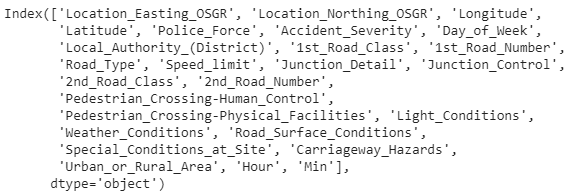


1. Sequence of predictive objectives

The prediction for accident severity of future accidents was the first objective since its prediction is independent of the outcome of other predictions and other predictions for future accidents rely on the severity of those accidents. The prediction for the number of casualties and the number of vehicles for future accidents were independent of each other but dependent on the accident severity. The prediction for the gender of the casualties depended upon the accident severity and the number of casualties. The prediction for the average age of the casualties depended upon the accident severity, the number of casualties and the gender of the casualties. For the machine learning algorithms, 70% of the data was used to train the model and 30% of the data was used to test the model.

### 4.3.1. Prediction for Accident Severity

The goal of this objective was to correctly identify the severity of an accident in future based on the information from the historical data. This was a classification task because the accident severity could possibly belong to any one of the severity class among fatal, serious or slight. The set of input variables considered for the prediction of accident severity can be seen in Figure 4.3.1.1.



1. Input variables for the prediction of accident severity

The machine learning algorithms tested for this task included Decision Tree Classifier, Random Forest Classifier and Naïve Bayes Classifier. As previously seen in Figure 4.2.1.1, the data had an imbalanced distribution of samples among the three severity classes. To address this problem, two methods called SMOTE and Random Under Sampler were tested. So, the experiment with the three models was carried out three times: first time with original data, second time with data having SMOTE applied to it and third time with data having Random Under Sampler applied to it. The count of training samples of each class after the application of the different technique is presented in Table 4.3.1.1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Technique applied to the data** | **Severity Class** | | |
| **Fatal** | **Serious** | **Slight** |
| **None** | 4732 | 60886 | 304754 |
| **SMOTE** | 304470 | 304235 | 304839 |
| **Random Under Sampler** | 4664 | 4621 | 4690 |

1. : Count of training samples of each severity class for different class imbalance handling techniques

The performance of all the models for every class imbalance handling technique is summarised in Table 4.3.1.2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Technique** | **Model** | **Severity Class** | **Performance Metric** | | | |
| **Precision** | **Recall** | **F1-score** | **Accuracy** |
| **None** | **Decision Tree** | Fatal | 0.03 | 0.03 | 0.03 | 69.80% |
| Serious | 0.19 | 0.21 | 0.2 |
| Slight | 0.83 | 0.81 | 0.82 |
| **Random Forest** | Fatal | 0.19 | 0 | 0.01 | 81.80% |
| Serious | 0.33 | 0.02 | 0.03 |
| Slight | 0.82 | 0.99 | 0.9 |
| **Naïve Bayes** | Fatal | 0.05 | 0.04 | 0.04 | 81.20% |
| Serious | 0.21 | 0 | 0.01 |
| Slight | 0.82 | 0.99 | 0.9 |
| **SMOTE** | **Decision Tree** | Fatal | 0.89 | 0.93 | 0.91 | 82.50% |
| Serious | 0.76 | 0.74 | 0.75 |
| Slight | 0.82 | 0.8 | 0.81 |
| **Random Forest** | Fatal | 0.97 | 0.98 | 0.97 | 91.20% |
| Serious | 0.97 | 0.78 | 0.86 |
| Slight | 0.83 | 0.98 | 0.9 |
| **Naïve Bayes** | Fatal | 0.46 | 0.71 | 0.56 | 46.20% |
| Serious | 0.42 | 0.33 | 0.37 |
| Slight | 0.51 | 0.35 | 0.41 |
| **Random Under Sampler** | **Decision Tree** | Fatal | 0.43 | 0.44 | 0.43 | 39.05% |
| Serious | 0.36 | 0.34 | 0.35 |
| Slight | 0.38 | 0.39 | 0.38 |
| **Random Forest** | Fatal | 0.52 | 0.58 | 0.55 | 45.54% |
| Serious | 0.38 | 0.32 | 0.35 |
| Slight | 0.45 | 0.47 | 0.46 |
| **Naïve Bayes** | Fatal | 0.48 | 0.62 | 0.54 | 44.05% |
| Serious | 0.39 | 0.09 | 0.15 |
| Slight | 0.41 | 0.62 | 0.5 |

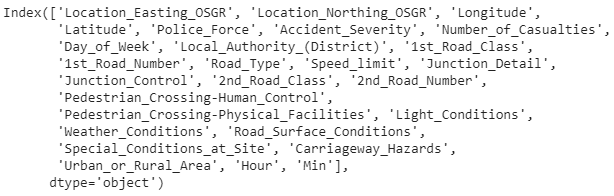
1. : Summary of machine learning models for prediction of accident severity

For this predictive task, it was crucial to reduce the number of false negatives, especially for ‘Fatal’ severity class, rather than reducing the number of false positives. Here false negatives meant the number of times the model predicted the severity to be ‘Slight’ or ‘Serious’ when actual class was ‘Fatal’, and false positives meant the number of times the model predicted the severity to be ‘Fatal’ when the actual class was ‘Serious’ or ‘Slight’. Hence, recall was chosen as the primary metric and was given more consideration than precision for the performance evaluation of the models.

From the above table, the base models, without any imbalance handling technique, performs extremely well for the prediction of ‘Slight’ severity class. This was due to a large number of samples for that class in the training set. However, the precision, recall and f1-score for other severity classes were extremely poor since the number of samples was less. After the application of SMOTE, a significant improvement in each model’s performance for the prediction of ‘Fatal’ and ‘Serious’ classes was observed. Random Forest worked exceptionally well with this technique. Random undersampling also performed better than the base model for the prediction of ‘Fatal’ and ‘Serious’ classes but it was not better than SMOTE.

### 4.3.2. Prediction for number of casualties

The goal of this objective was to correctly predict the number of casualties for an accident in future based on the information from the historical data. This was a regression task because the number of casualties was a continuous variable and could take any number, but it should be an integer value. The initial set of input variables considered for the prediction of the number of casualties can be seen in Figure 4.3.2.1.



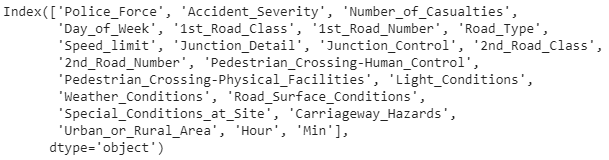
1. Initial set of input variables for the prediction of number of casualties

In regression tasks, it is important to check the multicollinearity between the input variables. This means that input variables should not be correlated or dependent on other input variables. Since in the case of multicollinearity, one input variable can be predicted using other input variables, removal of one of those input variables helps in removing this problem. Multicollinearity can be detected through the calculation of variation inflation factor (VIF). The VIF value for each column in the dataframe can be seen in Table 4.3.2.1.

|  |  |  |
| --- | --- | --- |
|  | Variables | VIF |
| 1 | Location\_Easting\_OSGR | 30173.63064 |
| 2 | Location\_Northing\_OSGR | 136.346064 |
| 3 | Longitude | 2437.650905 |
| 4 | Latitude | 46010.03385 |
| 5 | Police\_Force | 85.639861 |
| 6 | Accident\_Severity | 45.978628 |
| 7 | Number\_of\_Casualties | 4.012411 |
| 8 | Day\_of\_Week | 5.571848 |
| 9 | Local\_Authority\_(District) | 94.566499 |
| 10 | 1st\_Road\_Class | 12.710709 |
| 11 | 1st\_Road\_Number | 1.407064 |
| 12 | Road\_Type | 12.711698 |
| 13 | Speed\_limit | 18.861335 |
| 14 | Junction\_Detail | 4.005001 |
| 15 | Junction\_Control | 10.723787 |
| 16 | 2nd\_Road\_Class | 13.134478 |
| 17 | 2nd\_Road\_Number | 1.198069 |
| 18 | Pedestrian\_Crossing-Human\_Control | 1.044743 |
| 19 | Pedestrian\_Crossing-Physical\_Facilities | 1.400137 |
| 20 | Light\_Conditions | 2.650154 |
| 21 | Weather\_Conditions | 1.928517 |
| 22 | Road\_Surface\_Conditions | 6.214958 |
| 23 | Special\_Conditions\_at\_Site | 1.031763 |
| 24 | Carriageway\_Hazards | 1.022614 |
| 25 | Urban\_or\_Rural\_Area | 17.887745 |
| 26 | Hour | 8.473555 |

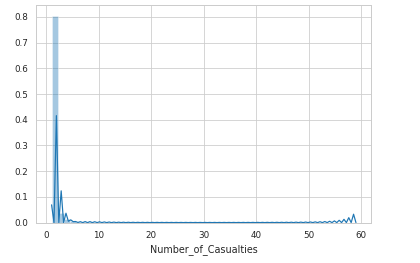
1. : VIF value of the columns for the prediction of number of casualties

The higher the value of VIF, the greater the multicollinearity. The set of input variables after removal of columns to deal with multicollinearity can be seen in Figure 4.3.2.2.



1. Final set of input variables for the prediction of number of casualties

After the removal of the columns, the columns containing label encoded data were converted to dummy encoded data to avoid bias towards those categories that were represented by larger numbers. The machine learning algorithms tested for this task included Linear Regression using Ordinary Least Square (OLS), Random Forest Regressor and K-Nearest Neighbour (KNN) Regressor. Before going ahead with the modelling, the distribution plot of the target variable i.e., the number of casualties was checked to understand the shape of the variable along with the descriptive statistics of the target variable.



1. Distribution plot of number of casualties

|  |  |
| --- | --- |
| **Measure** | **Value** |
| Count | 529103 |
| Mean | 1.320924 |
| Standard Deviation | 0.779283 |
| Min | 1 |
| 1st Quartile | 1 |
| 2nd Quartile | 1 |
| 3rd Quartile | 1 |
| Max | 59 |

1. : Descriptive statistics of the number of casualties

The distribution of the number of casualties was highly skewed with a maximum number of samples having a value of 1. This indicated a potential problem that the algorithm will predict the number of casualties in a future accident to be 1 in most cases.

For the selection of an optimal value of ‘k’ (number of neighbours used for prediction), the RMSE value for different values of ‘k’ was calculated and compared as shown in Table 4.3.2.3.

|  |  |
| --- | --- |
| **k value** | **RMSE** |
| 50 | 0.781155127 |
| 100 | 0.778228433 |
| 150 | 0.777346156 |
| 200 | 0.776863742 |
| 250 | 0.776679612 |
| 300 | 0.776583818 |
| 350 | 0.776672727 |
| 400 | 0.776592498 |
| 450 | 0.776583639 |
| 500 | 0.77662682 |
| 550 | 0.776666789 |
| 600 | 0.776725247 |

1. : RMSE for different values of k for prediction of number of casualties

From the above table, it can be seen that the root mean square error is minimum for the k value of 450. Thus, k=450 was selected for the KNN model.

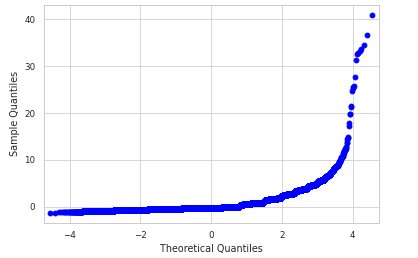
As discussed in chapter 3, RMSE and R2 score was used as the metric for the evaluation and selection of the suitable model for this task. For linear regression model, apart from RMSE and R2 score, the residuals i.e., the distance between the data point and the regression line, also needs to be evaluated in terms of its normality and homoscedasticity. The performance of all the models is summarised in Table 4.3.2.4.

|  |  |  |
| --- | --- | --- |
| Model | RMSE | R2 score |
| Linear Regression (OLS model) | 0.77 | 0.042 |
| Random Forest Regressor | 0.8 | -0.045 |
| KNN (k = 450) | 0.77 | 0.027 |

1. : Comparison of different models for the prediction of number of casualties

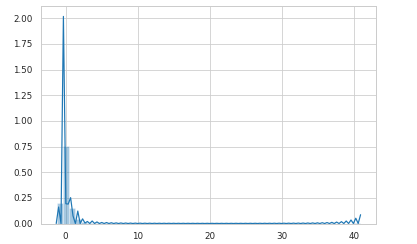
All the models had the RMSE value similar to the standard deviation of the target variable which indicated that most of the predictions were equal to the mean value of the target. The same thing was also seen in the R2 score of the models which were close to 0, suggesting that the models were preforming similar to predicting all future values as the mean value of the target variable.

The analysis of residuals for the linear regression model was also carried out to check whether the model was a good fit for the data or not. The analysis made use of Q-Q plot, residual distribution plot and fitted vs residuals plot as seen below.

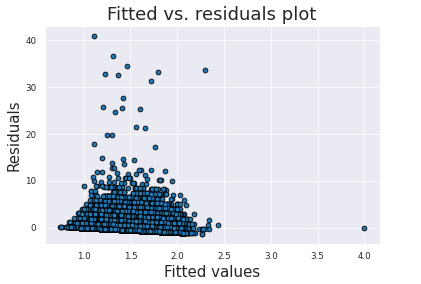


1. Q-Q plot of residuals for the prediction of number of casualties

The quantile-quantile (Q-Q) plot of the residuals did not follow a straight line along the diagonal which indicated that the residuals did not follow a normal distribution. Thus, the model was not appropriate for the task of prediction of the number of casualties. The shape of residual’s distribution was again confirmed with the help of a distribution plot as shown in Figure 4.3.2.5.



1. Distribution of residuals for the prediction of number of casualties

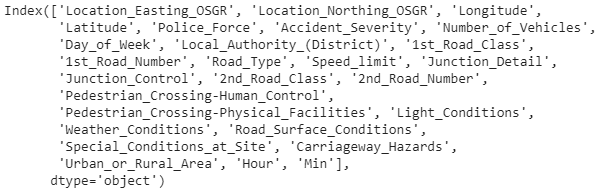


1. Fitted vs residuals plot for the prediction of number of casualties

The fitted vs residuals plot was plotted to check the homoscedasticity of the residuals. The residuals were found to be clustered together instead of being randomly separated which also suggested that the model was not a good fit. Thus, none of the models did much better than simply predicting all future values equal to the mean of the target column. However, if one model had to be chosen out of all, it would be linear regression since it had the best R2 score out of the three models.

### 4.3.3. Prediction for Number of Vehicles

The goal of this objective was to correctly predict the number of vehicles for an accident in future based on the information from the historical data. This was a regression task because the number of vehicles was a continuous variable and could take any number, but it should be an integer value. The initial set of input variables considered for the prediction of the number of vehicles can be seen in Figure 4.3.3.1.



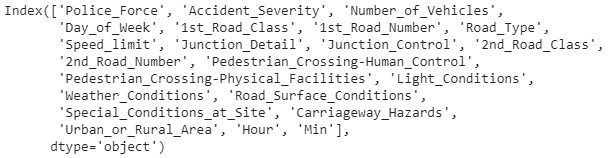
1. Initial set of input variables for the prediction of number of vehicles

Like the previous objective, the VIF of each column was calculated to check for the issue of multicollinearity. The VIF value for each column in the dataframe can be seen in Table 4.3.3.1.

|  |  |  |
| --- | --- | --- |
|  | Variables | VIF |
| 1 | Location\_Easting\_OSGR | 30177.3 |
| 2 | Location\_Northing\_OSGR | 136.437 |
| 3 | Longitude | 2438.02 |
| 4 | Latitude | 46029.9 |
| 5 | Police\_Force | 85.6192 |
| 6 | Accident\_Severity | 46.0862 |
| 7 | Number\_of\_Vehicles | 8.05011 |
| 8 | Day\_of\_Week | 5.57185 |
| 9 | Local\_Authority\_(District) | 94.5631 |
| 10 | 1st\_Road\_Class | 12.8397 |
| 11 | 1st\_Road\_Number | 1.40778 |
| 12 | Road\_Type | 12.7242 |
| 13 | Speed\_limit | 18.8068 |
| 14 | Junction\_Detail | 4.00623 |
| 15 | Junction\_Control | 10.7193 |
| 16 | 2nd\_Road\_Class | 13.1354 |
| 17 | 2nd\_Road\_Number | 1.19811 |
| 18 | Pedestrian\_Crossing-Human\_Control | 1.04473 |
| 19 | Pedestrian\_Crossing-Physical\_Facilities | 1.40651 |
| 20 | Light\_Conditions | 2.66103 |
| 21 | Weather\_Conditions | 1.92913 |
| 22 | Road\_Surface\_Conditions | 6.21958 |
| 23 | Special\_Conditions\_at\_Site | 1.03188 |
| 24 | Carriageway\_Hazards | 1.02363 |
| 25 | Urban\_or\_Rural\_Area | 17.8783 |
| 26 | Hour | 8.47305 |

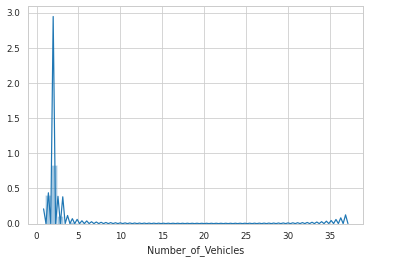
1. : VIF value of the columns for the prediction of number of vehicles

The set of input variables after removal of columns to deal with multicollinearity can be seen in Figure 4.3.3.2.



1. Final set of input variables for the prediction of number of vehicles

After the removal of the columns, the columns containing label encoded data were converted to dummy encoded data. The machine learning algorithms tested for this task included Linear Regression using OLS, Random Forest Regressor and KNN Regressor. Before going ahead with the modelling, the distribution plot of the target variable i.e., the number of vehicles was checked to understand the shape of the variable along with the descriptive statistics of the target variable.



1. Distribution plot of number of vehicles

|  |  |
| --- | --- |
| **Measure** | **Value** |
| Count | 529103 |
| Mean | 1.843382 |
| Standard Deviation | 0.715782 |
| Min | 1 |
| 1st Quartile | 1 |
| 2nd Quartile | 2 |
| 3rd Quartile | 2 |
| Max | 37 |

1. : Descriptive statistics of the number of vehicles

The distribution of the number of vehicles was also highly skewed with a maximum number of samples having a value of 2. This indicated a potential problem that the algorithm will predict the number of casualties in a future accident to be 2 in most cases.

For the selection of an optimal value of ‘k’ (number of neighbours used for prediction), the RMSE value for different values of ‘k’ was calculated and compared as shown in Table 4.3.3.3.

|  |  |
| --- | --- |
| **k value** | **RMSE** |
| 50 | 0.701124692 |
| 100 | 0.699742742 |
| 150 | 0.699706373 |
| 200 | 0.699946180 |
| 250 | 0.700206353 |
| 300 | 0.700504359 |

1. : RMSE for different values of k for prediction of number of vehicles

From the above table, it can be seen that the root mean square error is minimum for the k value of 150. Thus, k=150 was selected for the KNN model.

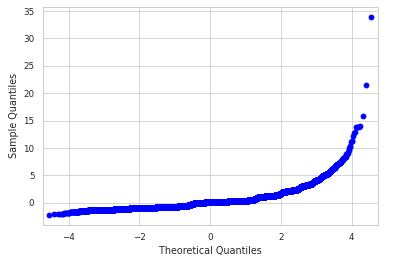
For this task also, RMSE and R2 score was used as the metric for the evaluation and selection of the suitable model. The performance of all the models is summarised in Table 4.3.3.4.

|  |  |  |
| --- | --- | --- |
| Model | RMSE | R2 score |
| Linear Regression (OLS model) | 0.68 | 0.074 |
| Random Forest Regressor | 0.7 | 0.037 |
| KNN (k = 150) | 0.69 | 0.045 |

1. : Comparison of different models for the prediction of number of vehicles

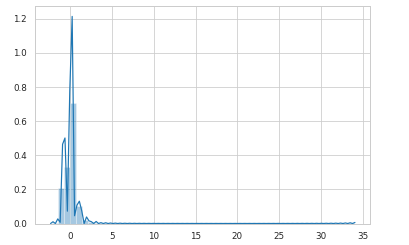
All the models had the RMSE value similar to the standard deviation of the target variable which indicated that most of the predictions were equal to the mean value of the target. The same thing was also seen in the R2 score of the models which were close to 0, suggesting that the models were preforming similar to predicting all future values as the mean value of the target variable.

Like the previous task, the analysis of residuals for the linear regression model was also carried out to check whether the model was a good fit for the data or not.

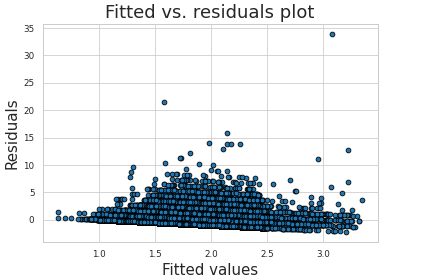


1. Q-Q plot of residuals for the prediction of number of vehicles

The Q-Q plot of the residuals did not follow a straight line along the diagonal which indicated that the residuals did not follow a normal distribution. Thus, the model was not appropriate for the task of prediction of the number of vehicles. The shape of residual’s distribution was again confirmed with the help of a distribution plot as shown in Figure 4.3.3.5.



1. Distribution of residuals for the prediction of number of vehicles

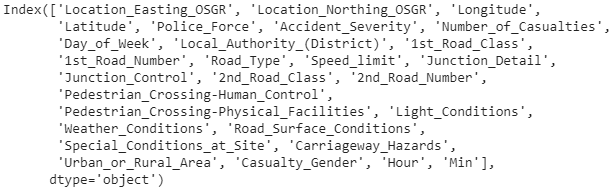


1. Fitted vs residuals plot for the prediction of number of vehicles

The fitted vs residuals plot showed that the residuals were clustered together instead of being randomly separated which also suggested that the model was not a good fit. Thus, none of the models did much better than simply predicting all future values equal to the mean of the target column. However, if one model had to be chosen out of all, it would be linear regression since it had the lowest RMSE and the best R2 score out of the three models.

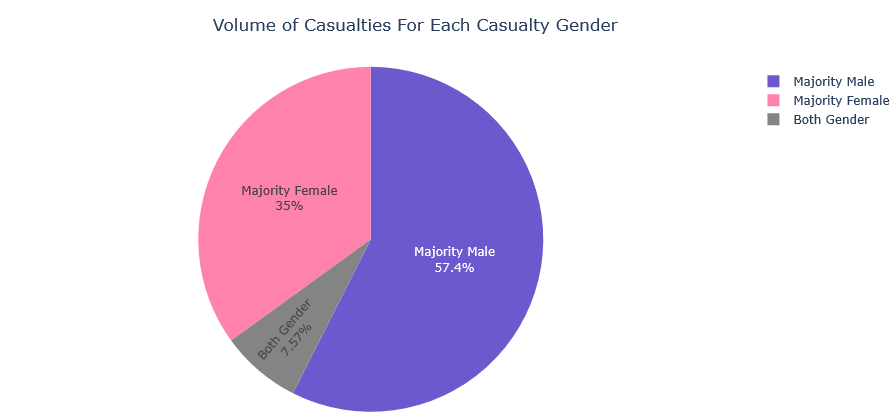
### 4.3.4. Prediction for Gender of Casualties

The goal of this objective was to correctly identify the most common gender of the casualties for an accident in future based on the information from the historical data. This was a classification task because the gender of the casualties could possibly belong to any one of the gender class among ‘Majority Male’, ‘Majority Female’ or ‘Both Gender’. The set of input variables considered for the prediction of the gender of the casualties can be seen in Figure 4.3.4.1.



1. Input variables for the prediction of the gender of casualties

Like the prediction of accident severity, the machine learning algorithms tested for this task included Decision Tree Classifier, Random Forest Classifier and Naïve Bayes Classifier.



1. Distribution of accidents for each gender class

As seen in Figure 4.3.4.2, the data had an imbalanced distribution of samples among the three gender classes. In this case also both SMOTE and Random Under Sampler were tested to resolve the issue of class imbalance. The count of training samples of each class after the application of the different technique is presented in Table 4.3.4.1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Technique applied to the data** | **Gender Class** | | |
| **Majority Male** | **Majority Female** | **Both Gender** |
| **None** | 212740 | 129691 | 27941 |
| **SMOTE** | 212593 | 213384 | 212332 |
| **Random Under Sampler** | 28120 | 27997 | 27981 |

1. : Count of training samples of each gender class for different class imbalance handling techniques

The performance of all the models for every class imbalance handling technique is summarised in Table 4.3.4.2.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Technique** | **Model** | **Gender Class** | **Performance Metric** | | | |
| **Precision** | **Recall** | **F1-score** | **Accuracy** |
| **None** | **Decision Tree** | Both Gender | 0.47 | 0.46 | 0.46 | 52.05% |
| Majority Female | 0.38 | 0.39 | 0.39 |
| Majority Male | 0.62 | 0.61 | 0.61 |
| **Random Forest** | Both Gender | 0.48 | 0.65 | 0.55 | 58.43% |
| Majority Female | 0.44 | 0.22 | 0.29 |
| Majority Male | 0.63 | 0.8 | 0.71 |
| **Naïve Bayes** | Both Gender | 0.15 | 0 | 0 | 55.99% |
| Majority Female | 0.38 | 0.1 | 0.15 |
| Majority Male | 0.58 | 0.92 | 0.71 |
| **SMOTE** | **Decision Tree** | Both Gender | 0.91 | 0.91 | 0.91 | 70.60% |
| Majority Female | 0.59 | 0.61 | 0.6 |
| Majority Male | 0.61 | 0.6 | 0.61 |
| **Random Forest** | Both Gender | 0.87 | 0.97 | 0.92 | 75.35% |
| Majority Female | 0.74 | 0.52 | 0.61 |
| Majority Male | 0.65 | 0.76 | 0.7 |
| **Naïve Bayes** | Both Gender | 0.42 | 0.67 | 0.52 | 41.93% |
| Majority Female | 0.39 | 0.29 | 0.34 |
| Majority Male | 0.49 | 0.25 | 0.35 |
| **Random Under Sampler** | **Decision Tree** | Both Gender | 0.83 | 0.79 | 0.81 | 58.76% |
| Majority Female | 0.47 | 0.47 | 0.47 |
| Majority Male | 0.48 | 0.5 | 0.49 |
| **Random Forest** | Both Gender | 0.83 | 0.99 | 0.9 | 65.99% |
| Majority Female | 0.54 | 0.49 | 0.51 |
| Majority Male | 0.55 | 0.5 | 0.52 |
| **Naïve Bayes** | Both Gender | 0.43 | 0.42 | 0.43 | 40.23% |
| Majority Female | 0.38 | 0.37 | 0.37 |
| Majority Male | 0.39 | 0.42 | 0.41 |

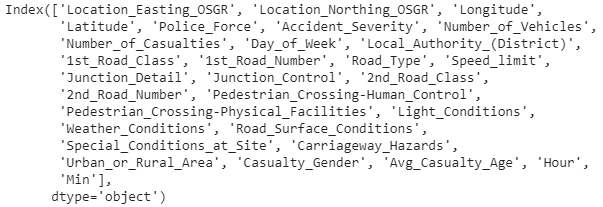
1. : Summary of machine learning models for prediction of gender of casualties

For this predictive task, neither precision nor recall was significantly important. Hence, the model accuracy was chosen as the primary metric since it indicates the general performance of the model and was given more consideration than precision or recall for the performance evaluation of the models.

From the above table, the base models, without any imbalance handling technique, performs well for the prediction of ‘Majority Male’ gender class. This was due to a large number of samples for that class in the training set. However, the precision, recall and f1-score for other gender classes were poor since the number of samples was less. After the application of SMOTE, a significant improvement in prediction for ‘Both Gender’ class along with a decent improvement in prediction for ‘Majority Female’ class in each model’s performance was observed. Random Under Sampler also displayed good improvement for the minority classes compared to the models using original data. Random Forest worked best for all the imbalance handling techniques well and particularly well when used with SMOTE since it achieved the best accuracy score out of all the models.

### 4.3.5. Prediction for Average Age of the Casualties

The goal of this objective was to correctly predict the average age of the casualties for an accident in future based on the information from the historical data. This was a regression task because the average age of casualties was a continuous variable and could take any number. The initial set of input variables considered for the prediction of the average age of the casualties can be seen in Figure 4.3.5.1.



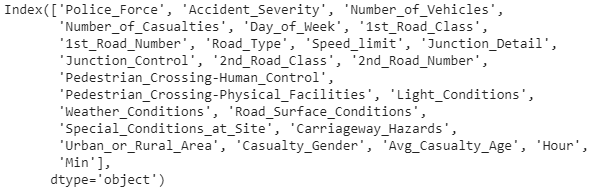
1. Initial set of input variables for the prediction of average age of casualties

Like the previous regression objectives, the VIF of each column was calculated to check for the issue of multicollinearity. Since VIF can only be calculated for numerical columns, the casualty gender column was dummy encoded before proceeding with VIF calculation. The VIF value for each column in the dataframe can be seen in Table 4.3.5.1.

|  |  |  |
| --- | --- | --- |
|  | Variables | VIF |
| 1 | Location\_Easting\_OSGR | 30050.5 |
| 2 | Location\_Northing\_OSGR | 136.773 |
| 3 | Longitude | 2447.08 |
| 4 | Latitude | 45945.7 |
| 5 | Police\_Force | 86.118 |
| 6 | Accident\_Severity | 46.8564 |
| 7 | Number\_of\_Vehicles | 8.54168 |
| 8 | Number\_of\_Casualties | 4.64947 |
| 9 | Day\_of\_Week | 5.56932 |
| 10 | Local\_Authority\_(District) | 95.1307 |
| 11 | 1st\_Road\_Class | 12.8481 |
| 12 | 1st\_Road\_Number | 1.40775 |
| 13 | Road\_Type | 12.7389 |
| 14 | Speed\_limit | 18.8988 |
| 15 | Junction\_Detail | 4.01096 |
| 16 | Junction\_Control | 10.818 |
| 17 | 2nd\_Road\_Class | 13.2429 |
| 18 | 2nd\_Road\_Number | 1.19872 |
| 19 | Pedestrian\_Crossing-Human\_Control | 1.0418 |
| 20 | Pedestrian\_Crossing-Physical\_Facilities | 1.4071 |
| 21 | Light\_Conditions | 2.69006 |
| 22 | Weather\_Conditions | 1.93757 |
| 23 | Road\_Surface\_Conditions | 6.26575 |
| 24 | Special\_Conditions\_at\_Site | 1.03172 |
| 25 | Carriageway\_Hazards | 1.02333 |
| 26 | Urban\_or\_Rural\_Area | 17.9515 |
| 27 | Avg\_Casualty\_Age | 5.4965 |
| 28 | Hour | 8.50779 |
| 29 | Min | 3.36966 |
| 30 | Casualty\_Gender\_Both gender | 1.2657 |
| 31 | Casualty\_Gender\_Majority Female | 1.65036 |

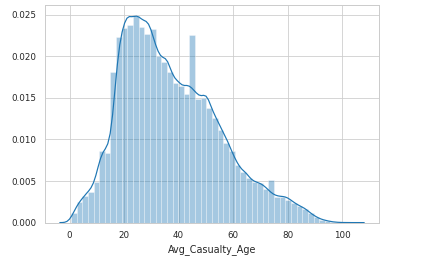
1. : VIF value of the columns for the prediction of average casualty age

The set of input variables after removal of columns to deal with multicollinearity can be seen in Figure 4.3.5.2.



1. Final set of input variables for the prediction of average casualty age

After the removal of the columns, the columns containing label encoded data were converted to dummy encoded data. The machine learning algorithms tested for this task included Linear Regression using OLS, Random Forest Regressor and KNN Regressor. Before going ahead with the modelling, the distribution plot of the target variable i.e., average casualty age was checked to understand the shape of the variable along with the descriptive statistics of the target variable.



1. Distribution plot of the average casualty age

|  |  |
| --- | --- |
| **Measure** | **Value** |
| Count | 522253 |
| Mean | 37.268 |
| Standard Deviation | 17.939 |
| Min | 0.500 |
| 1st Quartile | 23.500 |
| 2nd Quartile | 34.000 |
| 3rd Quartile | 49.000 |
| Max | 104.000 |

1. : Descriptive statistics of the average casualty age

The distribution of the average casualty age was also skewed but not as much as previous regression target variables, with the maximum number of samples between 20 to 35. This indicated that this task would not have the problem of predicting only mean value but rather it could have different prediction values for each test case.

For the selection of an optimal value of ‘k’ (number of neighbours used for prediction), the RMSE value for different values of ‘k’ was calculated and compared as shown in Table 4.3.5.3.

|  |  |
| --- | --- |
| **k value** | **RMSE** |
| 50 | 17.8540046 |
| 100 | 17.79909601 |
| 150 | 17.78694521 |
| 200 | 17.7879969 |
| 250 | 17.78825543 |
| 300 | 17.7924455 |

1. : RMSE for different values of k for prediction of average casualty age

From the above table, the root mean square error is minimum for the k value of 150. Thus, k=150 was selected for the KNN model.

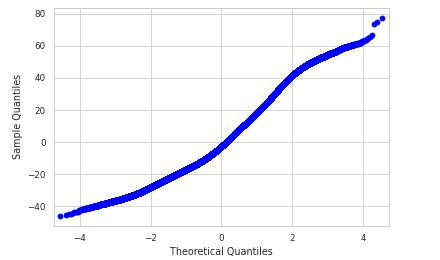
For this task also, RMSE and R2 score was used as the metric for the evaluation and selection of the suitable model. The performance of all the models is summarised in Table 4.3.5.4.

|  |  |  |
| --- | --- | --- |
| Model | RMSE | R2 score |
| Linear Regression (OLS model) | 17.62 | 0.034 |
| Random Forest Regressor | 17.55 | 0.042 |
| KNN (k = 150) | 17.78 | 0.016 |

1. : Comparison of different models for the prediction of average casualty age

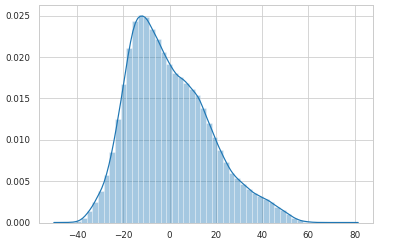
All the models had the RMSE value close to the standard deviation of the target variable which indicated that most of the predictions were like the mean value of the target. The same thing was also seen in the R2 score of the models which were close to 0, suggesting that the models were preforming like predicting all future values as the mean value of the target variable. This revealed that a good relationship was difficult to establish between the independent variables and the dependent variable.

Like the previous task, the analysis of residuals for the linear regression model was also carried out to check whether the model was a good fit for the data or not.

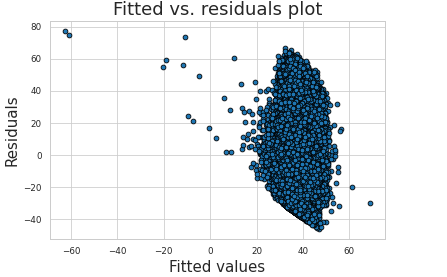


1. Q-Q plot of residuals for the prediction of average casualty age

The Q-Q plot of the residuals almost followed a straight line along the diagonal which indicated that the residuals somewhat followed a normal distribution. Thus, the model was appropriate for the task of prediction of the average age of casualties. The shape of residual’s distribution was again confirmed with the help of a distribution plot as shown in Figure 4.3.5.5.



1. Distribution of residuals for the prediction of average casualty age



1. Fitted vs residuals plot for the prediction of average casualty age

The fitted vs residuals plot showed that the residuals were clustered together instead of being randomly separated which also suggested that the model was not a good fit. Thus, none of the models did much better than simply predicting all future values equal to the mean of the target column. However, if one model had to be chosen out of all, it would be random forest regressor since it had the lowest RMSE and the best R2 score out of the three models.

To conclude, the machine learning algorithms performed better for the prediction of accident severity and the gender of the casualties which were the classification problems. On the flip side, for the regression tasks like the prediction of the number of casualties, the number of vehicles and the average age of the casualties, the machine learning algorithms did not perform significantly better than predicting the future values as the mean of the target variable.

# Chapter 5 – Discussion and Conclusions

## 5.1. Introduction

This chapter presents a discussion on the outcomes of the project and concludes the research project. The research aims and objectives of the project are revisited and the contribution of undertaken methods in the fulfilment of the objectives are assessed. The limitations of the project and the scope of future work are also discussed briefly in this chapter followed by the author’s reflection on the research project.

## 5.2. Discussion

This research project aimed to identify whether the historical data related to road safety and accidents for Great Britain help in predicting the likelihood of the severity of an accident and the details about the casualties for a specific location with particular circumstances. The work carried out in this project revealed that the historical data did help in predicting the severity of future accidents if the details regarding the location and circumstances are known. The majority gender of the casualties of an accident in future could also be predicted with good accuracy if other inputs are correct. However, the number of casualties was almost the same for all the future test cases and machine learning was not really required for the prediction.

The first objective of this project was to study the impact of different environmental and infrastructural factors on the accidents and identify the need for any infrastructural change. Along with that, the study was also aimed at spreading awareness in the public regarding these conditions. It was observed that accidents that occur in darkness with no artificial lighting have a higher tendency to be fatal than accidents that occur in well-lit areas. So, artificial lights need to be introduced in light deficit areas to reduce the fatality rate. Accidents in the presence of fog, mist or flood more than 3 cm deep were also more likely to be fatal than accidents in other weather conditions. Therefore, commuters should be cautious when travelling under these conditions. A defective road surface was found to be the most dangerous road condition with high chances of an accident being fatal. Thus, timely check-up and maintenance of road surface should be practised.

The second objective of this project was the visualization of accident hotspots on the map of Great Britain. The map seen in Figure 4.2.10.1 revealed that the southern part of Britain experienced a high volume of road accidents and most of the top five accident-prone districts and highways belonged to south England. Birmingham was found out to be the most affected region since both its district and highway local authorities were mentioned in fair share of the accidents.

The third and fourth objectives of this project were associated with the predictions and involved the use of machine learning techniques for classification and regression to predict the details about future accidents including the severity, the number of casualties and vehicles, the gender of the casualties and the average age of the casualties. The classification tasks suffered from the issue of class imbalance and to resolve that issue, SMOTE and Random Under Sampler techniques were employed and tested. Random Forest Classifier with minority oversampling using SMOTE worked best for the prediction of accident severity and gender of the casualties. The predictions for the number of casualties and the number of vehicles suffered from the issue of data skewness and thus the regression algorithms did not work much better than simply predicting the future values as the mean of the target variable for most of the test cases. The regression models for the prediction of the average age of the casualties performed comparatively better since the distribution of target variable resembled normal distribution. Despite the data being fit for regression, the performance of models was still not significantly better which indicated the lack of dependability of target variable on the input variables.

The fifth objective of this project was to suggest future work which can be carried out based on the findings of this project. Scope for future work regarding this project is repeating the research using the data from other parts of the world to check if the findings of this project apply globally. If the author could start the project afresh, he wished to monitor the trend of accidents over the years and forecast the course of accidents in the future. If the changes suggested by the author are implemented, then this research work could also be repeated in future to gauge the change in accident statistics after the implementation.

The analysis also revealed that most of the accidents belonged to slight severity and only 1.3% of accidents were fatal. However, this does not mean that these numbers cannot be reduced further. This research identified the factors present during most of the fatal accidents and the recommendations made during the analysis could prove to be helpful. Drivers or riders comprised majority of the casualties. Therefore, additional equipment for their safety should be introduced. The analysis in this research discovered that people aged between 26 to 35 years make up the highest number of casualties, but the literature review conducted for this project showed that most of the casualties in 2016 were aged between 17 to 24 years. This signifies that the trend has changed in recent times since the data considered for this project is newer.

## 5.3. Conclusion and Reflection

In conclusion, the project was able to successfully achieve the outlined objectives. The analysis of various factors related to accident location revealed the areas where more work needs to be done from an infrastructural point of view as well as public awareness point of view. The predictions for accident severity and gender of the casualties were accomplished successfully. The predictions for the number of casualties and the number of vehicles were difficult to realize since the target variable was mainly comprised of same value for the majority of the samples. The prediction for the average age of casualties was also not accomplished satisfactorily since there seemed to be a lack of relationship between the independent and the dependent variables.

The limitations of this research project include the unavailability of the data related to those accidents that were not reported to the police such as accidents on private property or parking lots. Since the data used for this project belonged to the United Kingdom, the findings of this work are limited only to the UK unless the research is repeated using the data from other parts of the world. The predictive tasks that aimed to predict the exact count or value of target variables (regression tasks) did not meet the expectation as they simply predicted the majority of future values as the mean of the target variable in most of the test cases.

Overall, this project was a great learning experience which involved all the aspects of a typical data analytics project. Some challenges faced during the completion of the project involved the presence of mixed data type in the accident index which resulted in the creation of incorrect data for models but was rectified later by converting the entire column to string data type. The data preparation phase, especially the creation of casualty gender and average casualty age columns from the casualty data for each accident index, took much longer time than expected since the computation had to be carried out for each iteration of the accident index. This project aimed to predict the exact severity class out of ‘Fatal’, ‘Serious’ or ‘Slight’ whereas, previous work done in this regard only classified accidents as fatal or non-fatal. The regression objectives of this project were also not seen in any work before this project based on the literature review conducted by the author. However, the findings of this research project suggest that the reason these objectives were not attempted before is due to the lack of relationship between the target and the input variables.

# References

Abrams, D. R. (n.d.) *DSS - Introduction to Regression*. Available at: https://dss.princeton.edu/online\_help/analysis/regression\_intro.htm (Accessed: 5 July 2020).

Aggarwal, C. C. (2015) ‘Data Classification’, in *Data Mining*. Cham: Springer International Publishing, pp. 285–344. doi: 10.1007/978-3-319-14142-8\_10.

Anaconda (n.d.) *Anaconda | Individual Edition*. Available at: https://www.anaconda.com/products/individual (Accessed: 24 July 2020).

Anderson, T. K. (2009) *Kernel density estimation and K-means clustering to profile road accident hotspots - ScienceDirect*, *Elsevier*. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0001457508002340 (Accessed: 22 May 2020).

Brownlee, J. (2020) *Random Oversampling and Undersampling for Imbalanced Classification*. Available at: https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/ (Accessed: 14 July 2020).

Chawla, N. V. *et al.* (2002) ‘SMOTE: Synthetic Minority Over-sampling Technique’, *Journal of Artificial Intelligence Research*, 2009(Sept. 28), pp. 321–357. doi: 10.1613/jair.953.

Connor, A. O. (2015) ‘An Analysis of the Predictive Capability of C5 . 0 and Chaid Decision Trees and Bayes Net in the Classification of fatal Traffic Accidents in the UK “ An analysis of the predictive capability of C5 . 0 and Chaid decision trees and Bayes net in the classif’.

Data Science Project Management (n.d.) *CRISP-DM – Data Science Project Management*. Available at: http://www.datascience-pm.com/crisp-dm-2/ (Accessed: 20 June 2020).

Department of Transport UK (2019) *Road Safety Data - data.gov.uk*. Available at: https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data (Accessed: 9 June 2020).

Gallo, A. (2015) *A Refresher on Regression Analysis*. Available at: https://hbr.org/2015/11/a-refresher-on-regression-analysis (Accessed: 5 July 2020).

Gandhi, R. (2018) *Introduction to Machine Learning Algorithms: Linear Regression*. Available at: https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a (Accessed: 24 May 2020).

Halton, C. (2019) *Predictive Analytics Definition*. Available at: https://www.investopedia.com/terms/p/predictive-analytics.asp (Accessed: 3 July 2020).

Kabli, A., Bhowmik, T. and Eluru, N. (2020) *A Multivariate Approach For Modeling Driver Injury Severity By Body Region*, *Analytic Methods in Accident Research*. Elsevier Ltd. doi: 10.1016/j.amar.2020.100129.

Larose, D. T. (2015) *Data Mining and Predictive Analytics*. Wiley (Wiley Series on Methods and Applications in Data Mining). Available at: https://books.google.ie/books?id=0HJuBwAAQBAJ.

Manasson, A. (2019) *Why using CRISP-DM will make you a better Data Scientist | by Alexander Manasson | Towards Data Science*. Available at: https://towardsdatascience.com/why-using-crisp-dm-will-make-you-a-better-data-scientist-66efe5b72686 (Accessed: 8 July 2020).

Martin, K. G. (n.d.) *Assessing the Fit of Regression Models - The Analysis Factor*. Available at: https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/ (Accessed: 24 July 2020).

Nabi, J. (2018) *Machine Learning — Multiclass Classification with Imbalanced Dataset*. Available at: https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a (Accessed: 19 June 2020).

O’Reilly, D. *et al.* (1994) *The Value of Road Safety: UK Research on the Valuation of Preventing Non-Fatal Injuries on JSTOR*, *Journal of Transport Economics and Policy*. Available at: https://www.jstor.org/stable/20053023?read-now=1&seq=1#page\_scan\_tab\_contents (Accessed: 22 May 2020).

OECD/ITF (2015) ‘Why Does Road Safety Improve When Economic Times Are Hard?’, p. 227.

OECD Health Statistics (2019) ‘OECD Health Statistics 2019 Definitions , Sources and Methods Injuries in road traffic accidents’, (54).

Park, E. S. *et al.* (2020) ‘Bayesian analysis of multivariate crash counts using copulas’, *Accident Analysis and Prevention*. Elsevier, (December 2019), p. 105431. doi: 10.1016/j.aap.2019.105431.

PAT Research (n.d.) *What is Predictive Analytics ?* Available at: https://www.predictiveanalyticstoday.com/what-is-predictive-analytics/ (Accessed: 24 May 2020).

Pete, C. *et al.* (2000) ‘CRISP-DM 1.0’, *CRISP-DM Consortium*, p. 76.

Rajput, A. (n.d.) *Types of Sources of Data in Data Mining - GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/types-of-sources-of-data-in-data-mining/ (Accessed: 13 July 2020).

Ray, S. (2015) *Regression Techniques in Machine Learning*. Available at: https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/ (Accessed: 5 July 2020).

Robineau, D. (2019) ‘Reported road casualties in Great Britain: 2018 annual report’, *Department for Transport*, 1(1), p. 36. doi: 10.1007/s10549-016-3800-5.

Rosenfeld, J. (2017) *Improving Road Safety and Traffic Safety*. Available at: https://www.natlawreview.com/article/improving-roadway-designs-traffic-safety (Accessed: 27 June 2020).

ROSPA (2017) *Road Safety Factsheet: Great Britain*, *The Royal Society for the Prevention of Accidents*. Available at: http://who.int/violence\_injury\_prevention/road\_safety\_status/2013/country\_profiles/brazil.pdf?ua=1.

Saltz, J. S. (2015) ‘The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness’, in *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015*. Institute of Electrical and Electronics Engineers Inc., pp. 2066–2071. doi: 10.1109/BigData.2015.7363988.

Samuel, A. L. (1969) ‘Some studies in machine learning using the game of checkers. II-Recent progress’, *Annual Review in Automatic Programming*, 6(PART 1), pp. 1–36. doi: 10.1016/0066-4138(69)90004-4.

Saunders, M., Lewis, P. and Thornhill, A. (2007) *Research methods for business student*. Fourth. FT Prentice Hall. Available at: http://ezproxy.ait.ie/login?url=http://search.ebscohost.com/login.aspx?direct=true&AuthType=shib&db=cat06239a&AN=ait.15110&site=eds-live.

Scikit Learn (n.d.) *1.10. Decision Trees — scikit-learn 0.23.1 documentation*. Available at: https://scikit-learn.org/stable/modules/tree.html (Accessed: 23 July 2020).

Scikit Learn (n.d.) *1.9. Naive Bayes — scikit-learn 0.23.1 documentation*. Available at: https://scikit-learn.org/stable/modules/naive\_bayes.html (Accessed: 23 July 2020).

Shmueli, G. and Koppius, O. R. (2011) ‘Predictive Analytics in Information Systems Research’, *MIS Quarterly*, 35(3), pp. 553–572.

Shung, K. P. (2018) *Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science*. Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 (Accessed: 24 July 2020).

Smart Vision - Europe (n.d.) *What is the CRISP-DM methodology?* Available at: https://www.sv-europe.com/crisp-dm-methodology/ (Accessed: 24 May 2020).

Stainton, H. (2020) *The research onion for beginners - Tourism Teacher.* Available at: https://tourismteacher.com/research-onion-beginners (Accessed: 24 July 2020).

Taylor, J. (2017) *Four Problems in Using CRISP-DM and How To Fix Them*. Available at: https://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html (Accessed: 8 July 2020).

Twin, A. (2019) *Data Mining Definition*, *Investopedia*. Available at: https://www.investopedia.com/terms/d/datamining.asp (Accessed: 24 May 2020).

Waseem, M. (2019) *Classification In Machine Learning | Classification Algorithms | Edureka*. Available at: https://www.edureka.co/blog/classification-in-machine-learning/ (Accessed: 24 May 2020).

Wegman, F. *et al.* (2015) ‘Evidence-based and data-driven road safety management’, *IATSS Research*. Elsevier Ltd, 39(1), pp. 19–25. doi: 10.1016/j.iatssr.2015.04.001.

World Health Organization (2018) *Global status report on road safety 2018*. Available at: https://www.who.int/publications/i/item/global-status-report-on-road-safety-2018.

World Health Organization (2020) *Road traffic injuries*. Available at: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (Accessed: 27 June 2020).

### Glossary of terms

Classification The process of identifying the class of a target out of a fixed set of possible classes

DataFrame A structured representation of the data in a tabular format

Dummy Encoding The process of representing categories by 0s and 1s by adding a column for each category

Label Encoding The process of assigning a number to each category

One-Hot Encoding Same as dummy encoding but required 1 less column

Oversampling The process of increasing the number of samples of a class

Regression The process of identifying the exact value of a continuous target

Residual The vertical distance between the data point and the regression line

Undersampling The process of decreasing the number of samples of a class

### List of Abbreviations

CRISP-DM – Cross Industry Process for Data Mining

DfT – Department for Transport

GDP – Gross Domestic Product

IDE – Integrated Development Environment

ITF – International Transport Forum

KNN – K Nearest Neighbour

LSOA – Leaving Scene of an Accident

OCED – Organization for Economic Co-operation and Development

OLS – Ordinary Least Square

Q-Q – Quantile-Quantile

RMSE – Root Mean Squared Error

ROSPA – Royal Society for the Prevention of Accidents

SMOTE – Synthetic Minority Oversampling Technique

UK – United Kingdom

VIF – Variation Inflation Factor

WHO – World Health Organization

# Appendix

The entire code for the project:

# Import necessary python libraries

import pandas as pd

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

import plotly.graph\_objects as go

import seaborn as sns

import statistics as st

# Read the accident data from csv files for 2015 to 2018

accident\_15 = pd.read\_csv('Accidents\_2015.csv')

accident\_16 = pd.read\_csv('Accidents\_2016.csv')

accident\_17 = pd.read\_csv('Accidents\_2017.csv')

accident\_18 = pd.read\_csv('Accidents\_2018.csv')

#For analysis we will merge all the accident data from 2015 to 2018 into a single dataframe

accident = accident\_15.append([accident\_16, accident\_17, accident\_18]).reset\_index(drop = True)

accident.head(5)

# Check number of rows and columns in the accident dataframe

accident.shape

# Count of null values in each column

accident.isna().sum()

# To impute missing speed limit values, we will check the correlation between speed limit and different features related to roads

accident['Speed\_limit'].corr(accident['1st\_Road\_Class'])

accident['Speed\_limit'].corr(accident['Road\_Surface\_Conditions'])

accident['Speed\_limit'].corr(accident['2nd\_Road\_Class'])

accident['Speed\_limit'].corr(accident['Accident\_Severity'])

# The column 'LSOA\_of\_Accident\_Location' has data only for England and Wales so we will not include this in our analysis

accident = accident.drop(['LSOA\_of\_Accident\_Location'], axis = 1)

# The count of missing data is very low compare to the volume of entire data and no logical method was found for imputation

# So, we will remove all the missing values

accident = accident.dropna()

# Basic descriptive statistics for accident data

accident.describe()

# Read the casualty data from csv files for 2015 to 2018

casualty\_15 = pd.read\_csv('Casualties\_2015.csv')

casualty\_16 = pd.read\_csv('Casualties\_2016.csv')

casualty\_17 = pd.read\_csv('Casualties\_2017.csv')

casualty\_18 = pd.read\_csv('Casualties\_2018.csv')

# For analysis we will merge all the casualty data from 2015 to 2018 into a single dataframe

casualty = casualty\_15.append([casualty\_16, casualty\_17, casualty\_18]).reset\_index(drop = True)

casualty.head(5)

# Check number of rows and columns in the casualty dataframe

casualty.shape

# Count of null values in each column

casualty.isna().sum()

# Basic descriptive statistics for casualty data

casualty.describe()

# Read the vehicle data from csv files for 2015 to 2018

vehicle\_15 = pd.read\_csv('Vehicles\_2015.csv')

vehicle\_16 = pd.read\_csv('Vehicles\_2016.csv')

vehicle\_17 = pd.read\_csv('Vehicles\_2017.csv')

vehicle\_18 = pd.read\_csv('Vehicles\_2018.csv')

# For analysis we will merge all the vehicle data from 2015 to 2018 into a single dataframe

vehicle = vehicle\_15.append([vehicle\_16, vehicle\_17, vehicle\_18]).reset\_index(drop = True)

vehicle.head(5)

# Check number of rows and columns in the vehicle dataframe

vehicle.shape

# Count of null values in each column

vehicle.isna().sum()

# Proportion of accident severities

accident['Accident\_Severity'].value\_counts().plot(kind = 'pie', autopct='%1.1f%%', startangle=90, figsize=(5, 5), title = 'Accident Severity')

plt.legend(['3 - Slight', '2 - Serious', '1 - Fatal'], loc = 'best')

# Dataframe to plot count of accidents for each day in a week

day\_df = pd.DataFrame({'Day\_of\_Week':['Friday','Thursday','Wednesday','Tuesday','Monday','Saturday','Sunday'], 'Count':accident['Day\_of\_Week'].value\_counts()})

day\_df

day\_df = day\_df.reset\_index()

day\_df = day\_df.sort\_values(by = ['index'])

fig = go.Figure()

fig.add\_trace(go.Bar(

x=day\_df['Day\_of\_Week'],

y=day\_df['Count'],

name='Day\_Bar'

))

fig.update\_layout(title={'text':'Count of Accidents For Each Day of the Week',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Day of the Week',

yaxis\_title = 'Count of Accidents')

fig.show()

# Dataframe for fatal accidents (severity 1)

accident\_fatal = accident[accident['Accident\_Severity']==1]

# Dictionary of accident count for each light condition

light\_dict = dict(accident[accident['Light\_Conditions'] != -1]['Light\_Conditions'].value\_counts())

light\_dict

# Dictionary of fatal accident count for each light condition

light\_dict\_fatal = dict(accident\_fatal[accident\_fatal['Light\_Conditions'] != -1]['Light\_Conditions'].value\_counts())

light\_dict\_fatal

light\_df = pd.DataFrame(zip(light\_dict.keys(), list(light\_dict.values())), columns = ['Light\_Condition', 'Count'])

light\_df

light\_df = light\_df.sort\_values(by=['Light\_Condition'])

light\_df

# Add proper labels for label-encoded light condition

light\_df['Light\_Condition'] = light\_df['Light\_Condition'].replace(to\_replace=[1,4,5,6,7],

value = ['Daylight', 'Darkness - lights lit', 'Darkness - lights unlit', 'Darkness - no lighting', 'Darkness - lighting unknown'])

light\_df = light\_df.reset\_index(drop=True)

light\_df

# Bar chart for accident count under each light condition

col = ['#FFD700','#CD0000','#912CEE','#555555','#8B3A3A']

fig = go.Figure(data=[go.Bar(

x=light\_df['Light\_Condition'],

y=light\_df['Count'],

text=light\_df['Count'],

textposition='outside',

marker\_color=col

)])

fig.update\_layout(title={'text':'Total Count of Accidents For Each Light Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Light Condition',

yaxis\_title = 'Count of Accidents')

light\_fatal\_df = pd.DataFrame(zip(light\_dict\_fatal.keys(), list(light\_dict\_fatal.values())),

columns = ['Light\_Condition', 'Count'])

light\_fatal\_df

light\_fatal\_df = light\_fatal\_df.sort\_values(by=['Light\_Condition'])

light\_fatal\_df

# Add proper labels for label-encoded light condition

light\_fatal\_df = light\_fatal\_df.replace(to\_replace=[1,4,5,6,7],

value = ['Daylight', 'Darkness - lights lit', 'Darkness - lights unlit', 'Darkness - no lighting',

'Darkness - lighting unknown'])

light\_fatal\_df = light\_fatal\_df.reset\_index(drop=True)

light\_fatal\_df

# Bar chart for fatal accident count under each light condition

col = ['#FFD700','#CD0000','#912CEE','#555555','#8B3A3A']

fig = go.Figure(data=[go.Bar(

x=light\_fatal\_df['Light\_Condition'],

y=light\_fatal\_df['Count'],

text=light\_fatal\_df['Count'],

textposition='outside',

marker\_color=col

)])

fig.update\_layout(title={'text':'Count of Fatal Accidents For Each Light Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Light Condition',

yaxis\_title = 'Count of Accidents')

light\_final\_df = pd.DataFrame()

light\_final\_df['Light\_Condition'] = light\_df['Light\_Condition']

light\_final\_df['Accident\_Count\_Total'] = light\_df['Count']

light\_final\_df['Accident\_Count\_Fatal'] = light\_fatal\_df['Count']

light\_final\_df['Fatal\_Proportion(%)'] = (light\_final\_df['Accident\_Count\_Fatal']/light\_final\_df['Accident\_Count\_Total'])\*100

light\_final\_df

# Bar chart for proportion of accidents that were fatal under each light condition

col = ['#FFD700','#CD0000','#912CEE','#555555','#8B3A3A']

fig = go.Figure(data=[go.Bar(

x=light\_final\_df['Light\_Condition'],

y=light\_final\_df['Fatal\_Proportion(%)'],

marker\_color=col

)])

fig.update\_layout(title={'text':'Proportion of Accidents that were Fatal Under Each Light Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Light Condition',

yaxis\_title = 'Percentage of Fatal Accidents (%)')

# Dictionary of accident count for each weather condition

weather\_dict = dict(accident[accident['Weather\_Conditions'] != -1]['Weather\_Conditions'].value\_counts())

weather\_dict

# Dictionary of fatal accident count for each weather condition

weather\_dict\_fatal = dict(accident\_fatal[accident\_fatal['Weather\_Conditions'] != -1]['Weather\_Conditions'].value\_counts())

weather\_dict\_fatal

weather\_df = pd.DataFrame(zip(weather\_dict.keys(), list(weather\_dict.values())), columns = ['Weather\_Condition', 'Count'])

weather\_df

weather\_df = weather\_df.sort\_values(by=['Weather\_Condition'])

weather\_df

# Add proper labels for label-encoded weather condition

weather\_df = weather\_df.replace(to\_replace=[1,2,3,4,5,6,7,8,9],

value = ['Fine no high winds', 'Raining no high winds', 'Snowing no high winds', 'Fine + high winds',

'Raining + high winds', 'Snowing + high winds', 'Fog or mist', 'Other', 'Unknown'])

weather\_df = weather\_df.reset\_index(drop = True)

weather\_df

# Bar chart for accident count under each weather condition

col = ['#CD0000','#27408B','#71C671','#8E388E','#FF8000','#F0E68C','#CD661D','#FF1493','#696969']

fig = go.Figure(data=[go.Bar(

x=weather\_df['Weather\_Condition'],

y=weather\_df['Count'],

marker\_color=col

)])

fig.update\_layout(title={'text':'Total Count of Accidents For Each Weather Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Weather Condition',

yaxis\_title = 'Count of Accidents')

weather\_fatal\_df = pd.DataFrame(zip(weather\_dict\_fatal.keys(), list(weather\_dict\_fatal.values())),

columns = ['Weather\_Condition', 'Count'])

weather\_fatal\_df

weather\_fatal\_df = weather\_fatal\_df.sort\_values(by=['Weather\_Condition'])

weather\_fatal\_df

# Add proper labels for label-encoded weather condition

weather\_fatal\_df['Weather\_Condition'] = weather\_fatal\_df['Weather\_Condition'].replace(to\_replace=[1,2,3,4,5,6,7,8,9],

value = ['Fine no high winds', 'Raining no high winds', 'Snowing no high winds', 'Fine + high winds',

'Raining + high winds', 'Snowing + high winds', 'Fog or mist', 'Other', 'Unknown'])

weather\_fatal\_df = weather\_fatal\_df.reset\_index(drop=True)

weather\_fatal\_df

# Bar chart for fatal accident count under each weather condition

col = ['#CD0000','#27408B','#71C671','#8E388E','#FF8000','#F0E68C','#CD661D','#FF1493','#696969']

fig = go.Figure(data=[go.Bar(

x=weather\_fatal\_df['Weather\_Condition'],

y=weather\_fatal\_df['Count'],

marker\_color=col

)])

fig.update\_layout(title={'text':'Count of Fatal Accidents For Each Weather Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Weather Condition',

yaxis\_title = 'Count of Accidents')

weather\_final\_df = pd.DataFrame()

weather\_final\_df['Weather\_Condition'] = weather\_df['Weather\_Condition']

weather\_final\_df['Accident\_Count\_Total'] = weather\_df['Count']

weather\_final\_df['Accident\_Count\_Fatal'] = weather\_fatal\_df['Count']

weather\_final\_df['Fatal\_Proportion(%)'] = (weather\_final\_df['Accident\_Count\_Fatal']/weather\_final\_df['Accident\_Count\_Total'])\*100

weather\_final\_df

# Bar chart for proportion of accidents that were fatal under each weather condition

col = ['#CD0000','#27408B','#71C671','#8E388E','#FF8000','#F0E68C','#CD661D','#FF1493','#696969']

fig = go.Figure(data=[go.Bar(

x=weather\_final\_df['Weather\_Condition'],

y=weather\_final\_df['Fatal\_Proportion(%)'],

marker\_color=col

)])

fig.update\_layout(title={'text':'Proportion of Accidents that were Fatal Under Each Weather Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Weather Condition',

yaxis\_title = 'Percentage of Fatal Accidents (%)')

# Dictionary of accident count for each special condition

sc\_dict = dict(accident\_sc['Special\_Conditions\_at\_Site'].value\_counts())

sc\_dict

# Dictionary of fatal accident count for each special condition

sc\_dict\_fatal = dict(accident\_fatal\_sc['Special\_Conditions\_at\_Site'].value\_counts())

sc\_dict\_fatal

sc\_dict\_key = ['Auto traffic signal - out', 'Auto signal part defective', 'Road sign or marking defective or obscured',

'Roadworks', 'Road surface defective', 'Oil or diesel', 'Mud']

sc\_df = pd.DataFrame(zip(sc\_dict.keys(), list(sc\_dict.values())), columns = ['Special\_Conditions\_at\_Site', 'Count'])

sc\_df

sc\_df = sc\_df.sort\_values(by=['Special\_Conditions\_at\_Site'])

sc\_df

sc\_df['Special\_Conditions\_at\_Site'] = sc\_df['Special\_Conditions\_at\_Site'].replace(

to\_replace = [1,2,3,4,5,6,7],value = ['Auto traffic signal - out','Auto signal part defective',

'Road sign or marking defective or obscured','Roadworks', 'Road surface defective',

'Oil or diesel', 'Mud'])

sc\_df = sc\_df.reset\_index(drop=True)

sc\_df

# Bar chart for accident count under each special condition

clr = ['#CD0000','#27408B','#71C671','#8E388E','#FF8000','#F0E68C','#CD661D','#FF1493','#696969']

fig = go.Figure(data=[go.Bar(

x=sc\_df['Special\_Conditions\_at\_Site'],

y=sc\_df['Count'],

marker\_color=clr

)])

fig.update\_layout(title={'text':'Total Count of Accidents For Each Special Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Special Condition at Site',

yaxis\_title = 'Count of Accidents')

sc\_fatal\_df = pd.DataFrame(zip(sc\_dict\_fatal.keys(), list(sc\_dict\_fatal.values())),

columns = ['Special\_Conditions\_at\_Site', 'Count'])

sc\_fatal\_df

sc\_fatal\_df = sc\_fatal\_df.sort\_values(by=['Special\_Conditions\_at\_Site'])

sc\_fatal\_df

sc\_fatal\_df['Special\_Conditions\_at\_Site'] = sc\_fatal\_df['Special\_Conditions\_at\_Site'].replace(

to\_replace = [1,2,3,4,5,6,7],value = ['Auto traffic signal - out','Auto signal part defective',

'Road sign or marking defective or obscured','Roadworks', 'Road surface defective',

'Oil or diesel', 'Mud'])

sc\_fatal\_df = sc\_fatal\_df.reset\_index(drop=True)

sc\_fatal\_df

# Bar chart for fatal accident count under each special condition

clr = ['#CD0000','#27408B','#71C671','#8E388E','#FF8000','#F0E68C','#CD661D','#FF1493','#696969']

fig = go.Figure(data=[go.Bar(

x=sc\_fatal\_df['Special\_Conditions\_at\_Site'],

y=sc\_fatal\_df['Count'],

marker\_color=clr

)])

fig.update\_layout(title={'text':'Count of Fatal Accidents For Each Special Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Special Condition at Site',

yaxis\_title = 'Count of Accidents')

sc\_final\_df = pd.DataFrame()

sc\_final\_df['Special\_Conditions\_at\_Site'] = sc\_df['Special\_Conditions\_at\_Site']

sc\_final\_df['Accident\_Count\_Total'] = sc\_df['Count']

sc\_final\_df['Accident\_Count\_Fatal'] = sc\_fatal\_df['Count']

sc\_final\_df['Fatal\_Proportion(%)'] = (sc\_final\_df['Accident\_Count\_Fatal']/sc\_final\_df['Accident\_Count\_Total'])\*100

sc\_final\_df

# Bar chart for proportion of accidents that were fatal under each special condition

clr = ['#CD0000','#27408B','#71C671','#8E388E','#FF8000','#F0E68C','#CD661D','#FF1493','#696969']

fig = go.Figure(data=[go.Bar(

x=sc\_final\_df['Special\_Conditions\_at\_Site'],

y=sc\_final\_df['Fatal\_Proportion(%)'],

marker\_color=clr

)])

fig.update\_layout(title={'text':'Proportion of Accidents that were Fatal Under Each Special Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Special Condition at Site',

yaxis\_title = 'Percentage of Fatal Accidents (%)')

road\_dict = dict(accident[accident['Road\_Surface\_Conditions'] != -1]['Road\_Surface\_Conditions'].value\_counts())

road\_dict

road\_dict\_fatal = dict(accident\_fatal[accident\_fatal['Road\_Surface\_Conditions'] != -1]['Road\_Surface\_Conditions'].value\_counts())

road\_dict\_fatal

road\_df = pd.DataFrame(zip(road\_dict.keys(), list(road\_dict.values())), columns = ['Road\_Surface\_Conditions', 'Count'])

road\_df

road\_df = road\_df.sort\_values(by=['Road\_Surface\_Conditions'])

road\_df

road\_df['Road\_Surface\_Conditions'] = road\_df['Road\_Surface\_Conditions'].replace(

to\_replace = [1,2,3,4,5], value = ['Dry', 'Wet or damp', 'Snow', 'Frost or ice', 'Flood over 3cm. deep'])

road\_df = road\_df.reset\_index(drop=True)

road\_df

clr = ['#EEEE00','#008080','#C71585','#6959CD','#8B4726']

fig = go.Figure(data=[go.Bar(

x=road\_df['Road\_Surface\_Conditions'],

y=road\_df['Count'],

marker\_color=clr

)])

fig.update\_layout(title={'text':'Total Count of Accidents For Each Road Surface Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Road Surface Condition',

yaxis\_title = 'Count of Accidents')

road\_fatal\_df = pd.DataFrame(zip(road\_dict\_fatal.keys(), list(road\_dict\_fatal.values())),

columns = ['Road\_Surface\_Conditions', 'Count'])

road\_fatal\_df

road\_fatal\_df = road\_fatal\_df.sort\_values(by=['Road\_Surface\_Conditions'])

road\_fatal\_df

road\_fatal\_df['Road\_Surface\_Conditions'] = road\_fatal\_df['Road\_Surface\_Conditions'].replace(

to\_replace = [1,2,3,4,5], value = ['Dry', 'Wet or damp', 'Snow', 'Frost or ice', 'Flood over 3cm. deep'])

road\_fatal\_df = road\_fatal\_df.reset\_index(drop=True)

road\_fatal\_df

clr = ['#EEEE00','#008080','#C71585','#6959CD','#8B4726']

fig = go.Figure(data=[go.Bar(

x=road\_fatal\_df['Road\_Surface\_Conditions'],

y=road\_fatal\_df['Count'],

marker\_color=clr

)])

fig.update\_layout(title={'text':'Count of Fatal Accidents For Each Road Surface Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Road Surface Condition',

yaxis\_title = 'Count of Accidents')

road\_final\_df = pd.DataFrame()

road\_final\_df['Road\_Surface\_Conditions'] = road\_df['Road\_Surface\_Conditions']

road\_final\_df['Accident\_Count\_Total'] = road\_df['Count']

road\_final\_df['Accident\_Count\_Fatal'] = road\_fatal\_df['Count']

road\_final\_df['Fatal\_Proportion(%)'] = (road\_final\_df['Accident\_Count\_Fatal']/road\_final\_df['Accident\_Count\_Total'])\*100

road\_final\_df

clr = ['#EEEE00','#008080','#C71585','#6959CD','#8B4726']

fig = go.Figure(data=[go.Bar(

x=road\_final\_df['Road\_Surface\_Conditions'],

y=road\_final\_df['Fatal\_Proportion(%)'],

marker\_color=clr

)])

fig.update\_layout(title={'text':'Proportion of Accidents that were Fatal Under Each Road Surface Condition',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Road Surface Condition',

yaxis\_title = 'Percentage of Fatal Accidents (%)')

# To find count of accidents for each month, the data column needs to be split to get day, month and year

accident[['Day','Month','Year']] = accident['Date'].str.split('/', expand = True)

accident.head(5)

accident\_fatal =accident[accident['Accident\_Severity']==1]

month\_dict = dict(accident['Month'].value\_counts())

month\_dict

month\_df = pd.DataFrame(zip(month\_dict.keys(), list(month\_dict.values())),

columns = ['Month', 'Count'])

month\_df

month\_df = month\_df.sort\_values(by='Month')

month\_df

month\_df['Month'] = month\_df['Month'].replace(

to\_replace = ['11', '07', '10', '09', '06', '05', '01', '12', '08', '03', '04', '02']

,value = ['November','July','October','September','June','May','January','December','August','March','April','February'])

month\_df = month\_df.reset\_index(drop=True)

month\_df

fig = go.Figure(data=[go.Bar(

x=month\_df['Month'],

y=month\_df['Count'],

marker\_color = 'brown'

)])

fig.update\_layout(title={'text':'Total Count of Accidents For Each Month',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Month',

yaxis\_title = 'Count of Accidents')

month\_fatal\_dict = dict(accident\_fatal['Month'].value\_counts())

month\_fatal\_dict

month\_fatal\_df = pd.DataFrame(zip(month\_fatal\_dict.keys(), list(month\_fatal\_dict.values())),

columns = ['Month', 'Count'])

month\_fatal\_df

month\_fatal\_df = month\_fatal\_df.sort\_values(by='Month')

month\_fatal\_df

month\_fatal\_df['Month'] = month\_fatal\_df['Month'].replace(

to\_replace = ['11', '07', '10', '09', '06', '05', '01', '12', '08', '03', '04', '02']

,value = ['November','July','October','September','June','May','January','December','August','March','April','February'])

month\_fatal\_df = month\_fatal\_df.reset\_index(drop=True)

month\_fatal\_df

fig = go.Figure(data=[go.Bar(

x=month\_fatal\_df['Month'],

y=month\_fatal\_df['Count'],

marker\_color = 'brown'

)])

fig.update\_layout(title={'text':'Count of Fatal Accidents For Each Month',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Month',

yaxis\_title = 'Count of Accidents')

month\_final\_df = pd.DataFrame()

month\_final\_df['Month'] = month\_df['Month']

month\_final\_df['Accident\_Count\_Total'] = month\_df['Count']

month\_final\_df['Accident\_Count\_Fatal'] = month\_fatal\_df['Count']

month\_final\_df['Fatal\_Proportion(%)'] = (month\_final\_df['Accident\_Count\_Fatal']/month\_final\_df['Accident\_Count\_Total'])\*100

month\_final\_df

fig = go.Figure(data=[go.Bar(

x=month\_final\_df['Month'],

y=month\_final\_df['Fatal\_Proportion(%)'],

marker\_color='brown'

)])

fig.update\_layout(title={'text':'Proportion of Accidents that were Fatal Under Each Month',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Month',

yaxis\_title = 'Percentage of Fatal Accidents (%)')

# Remove newly added day, month and year columns

accident = accident.drop(['Day','Month','Year'], axis=1)

accident.head(5)

accident\_fatal = accident\_fatal.drop(['Day','Month','Year'], axis=1)

accident\_fatal.head(5)

# To find count of accidents for each hour, the time column needs to be split to get hour and minute

accident[['Hour','Min']] = accident['Time'].str.split(':', expand = True)

accident.head(5)

hr\_dict = dict(accident['Hour'].value\_counts())

hr\_dict

hr\_df = pd.DataFrame(zip(hr\_dict.keys(), list(hr\_dict.values())),

columns = ['Hour', 'Count'])

hr\_df

hr\_df = hr\_df.sort\_values(by=['Hour'])

hr\_df

hr\_df = hr\_df.reset\_index(drop=True)

hr\_df

fig = go.Figure(data=[go.Bar(

x=hr\_df['Hour'],

y=hr\_df['Count'],

marker\_color = 'orange'

)])

fig.update\_layout(title={'text':'Total Count of Accidents For Each Hour',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Hour',

yaxis\_title = 'Count of Accidents')

hr\_fatal\_df = pd.DataFrame(zip(hr\_fatal\_dict.keys(), list(hr\_fatal\_dict.values())),

columns = ['Hour', 'Count'])

hr\_fatal\_df

hr\_fatal\_df = hr\_fatal\_df.sort\_values(by=['Hour'])

hr\_fatal\_df

hr\_fatal\_df = hr\_fatal\_df.reset\_index(drop=True)

hr\_fatal\_df

fig = go.Figure(data=[go.Bar(

x=hr\_fatal\_df['Hour'],

y=hr\_fatal\_df['Count'],

marker\_color = 'orange'

)])

fig.update\_layout(title={'text':'Count of Fatal Accidents For Each Hour',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Hour',

yaxis\_title = 'Count of Accidents')

hr\_final\_df = pd.DataFrame()

hr\_final\_df['Hour'] = hr\_df['Hour']

hr\_final\_df['Accident\_Count\_Total'] = hr\_df['Count']

hr\_final\_df['Accident\_Count\_Fatal'] = hr\_fatal\_df['Count']

hr\_final\_df['Fatal\_Proportion(%)'] = (hr\_final\_df['Accident\_Count\_Fatal']/hr\_final\_df['Accident\_Count\_Total'])\*100

hr\_final\_df

fig = go.Figure(data=[go.Bar(

x=hr\_final\_df['Hour'],

y=hr\_final\_df['Fatal\_Proportion(%)'],

marker\_color='orange'

)])

fig.update\_layout(title={'text':'Proportion of Accidents that were Fatal Under Each Hour',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Hour',

yaxis\_title = 'Percentage of Fatal Accidents (%)')

#Top 5 districts in terms of number of accidents

top5\_district\_main = accident['Local\_Authority\_(District)'].value\_counts()[:5]

top5\_district\_main

#Convert the value counts into a dictionary for plotting

sns.set(style="darkgrid", palette="Set1")

dict\_top5\_district = dict(top5\_district\_main)

dict\_top5\_district

#district\_keys = ['Westminster', 'Lambeth', 'Leeds', 'Birmingham', 'Cornwall']

top5\_district = dict()

top5\_district['Westminster'] = dict\_top5\_district[1]

top5\_district['Lambeth'] = dict\_top5\_district[9]

top5\_district['Leeds'] = dict\_top5\_district[204]

top5\_district['Birmingham'] = dict\_top5\_district[300]

top5\_district['Cornwall'] = dict\_top5\_district[596]

top5\_district

clr = ['#1874CD','#6E8B3D','#8A2BE2','#EE6A50','#CDCD00']

fig = go.Figure(data=[go.Bar(

x=list(top5\_district.keys()),

y=list(top5\_district.values()),

marker\_color=clr

)])

fig.update\_layout(title={'text':'Count of Accidents for Top 5 Accident Prone Districts',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Local Authority (District)',

yaxis\_title = 'Count of Accidents')

#Top 5 highways in terms of number of accidents

top5\_highway\_main = accident['Local\_Authority\_(Highway)'].value\_counts()[:5]

top5\_highway\_main

#Convert the value counts into a dictionary for plotting

sns.set(style="darkgrid", palette="Set2")

dict\_top5\_highway = dict(top5\_highway\_main)

dict\_top5\_highway

top5\_highway = dict()

top5\_highway['Birmingham'] = dict\_top5\_highway['E08000025']

top5\_highway['Essex'] = dict\_top5\_highway['E10000012']

top5\_highway['Hampshire'] = dict\_top5\_highway['E10000014']

top5\_highway['Kent'] = dict\_top5\_highway['E10000016']

top5\_highway['Surrey'] = dict\_top5\_highway['E10000030']

top5\_highway

clr = ['#53868B','#9BCD9B','#CD6090','#03A89E','#FF8247']

fig = go.Figure(data=[go.Bar(

x=list(top5\_highway.keys()),

y=list(top5\_highway.values()),

marker\_color=clr

)])

fig.update\_layout(title={'text':'Count of Accidents for Top 5 Accident Prone Highways',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Local Authority (Highway)',

yaxis\_title = 'Count of Accidents')

# Dictionary for accident count under each propulsion code

prop\_dict = dict(vehicle['Propulsion\_Code'].value\_counts())

prop\_dict

propulsion\_dict = dict()

propulsion\_dict['Unknown'] = prop\_dict[-1]

propulsion\_dict['Petrol'] = prop\_dict[1]

propulsion\_dict['Heavy oil'] = prop\_dict[2]

propulsion\_dict['Electric'] = prop\_dict[3]

propulsion\_dict['Steam'] = prop\_dict[4]

propulsion\_dict['Gas'] = prop\_dict[5]

propulsion\_dict['Petrol/Gas (LPG)'] = prop\_dict[6]

propulsion\_dict['Gas/Bi-fuel'] = prop\_dict[7]

propulsion\_dict['Hybrid electric'] = prop\_dict[8]

propulsion\_dict['Gas Diesel'] = prop\_dict[9]

propulsion\_dict['New fuel technology'] = prop\_dict[10]

propulsion\_dict['Electric diesel'] = prop\_dict[12]

propulsion\_dict

clr = ['#555555','#EE2C2C','#8B6914','#458B00','#00CDCD','#EEAD0E','#556B2F','#CD6600','#00CD00','#4B0082','#CD6090','#778899']

fig = go.Figure(data=[go.Bar(

x=list(propulsion\_dict.keys()),

y=list(propulsion\_dict.values()),

marker\_color=clr

)])

fig.update\_layout(title={'text':'Count of Accidents for Each Propulsion Type',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Propulsion Type',

yaxis\_title = 'Count of Accidents')

# Merge vehicle and accident dataframe on accident index

vehicle\_new = vehicle.merge(accident, on = 'Accident\_Index')

#Average age of vehicles with petrol type propulsion in top 5 districts

avg\_vehicleAge\_district = {}

for dist, val in top5\_district\_main.items():

avg\_vehicleAge\_district[dist] = st.mean(

Vehicle\_new\_petrol[Vehicle\_new\_petrol['Local\_Authority\_(District)'] == dist]['Age\_of\_Vehicle'])

avg\_vehicleAge\_district2 = dict()

avg\_vehicleAge\_district2['Westminster'] = avg\_vehicleAge\_district[1]

avg\_vehicleAge\_district2['Lambeth'] = avg\_vehicleAge\_district[9]

avg\_vehicleAge\_district2['Leeds'] = avg\_vehicleAge\_district[204]

avg\_vehicleAge\_district2['Birmingham'] = avg\_vehicleAge\_district[300]

avg\_vehicleAge\_district2['Cornwall'] = avg\_vehicleAge\_district[596]

avg\_vehicleAge\_district2

clr = ['#1874CD','#6E8B3D','#8A2BE2','#EE6A50','#CDCD00']

fig = go.Figure(data=[go.Bar(

x=list(avg\_vehicleAge\_district2.keys()),

y=list(avg\_vehicleAge\_district2.values()),

marker\_color=clr

)])

fig.update\_layout(title={'text':'Average Age of Vehicles with Petrol Type Populsion in Top 5 Accident Prone Districts',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Local Authority (District)',

yaxis\_title = 'Average Vehicle Age')

# Dictionary for casualty count for each casualty class

casualty\_class\_dict = dict(casualty['Casualty\_Class'].value\_counts())

casualty\_class\_dict

casualty\_class\_df = pd.DataFrame(zip(casualty\_class\_dict.keys(), list(casualty\_class\_dict.values())),

columns = ['Casualty\_Class', 'Count'])

casualty\_class\_df

casualty\_class\_df['Casualty\_Class'] = casualty\_class\_df['Casualty\_Class'].replace(

to\_replace = [1,2,3], value = ['Driver or Rider','Passenger','Pedestrian'])

casualty\_class\_df

# Pie chart for distribution of casualties under each casualty class

clr = ['#FA8072','#63B8FF','#00FA9A']

fig = go.Figure()

fig.add\_trace(go.Pie(

labels=casualty\_class\_df['Casualty\_Class'],

values=casualty\_class\_df['Count'],

textinfo='label+percent',

name='Casualty\_Class\_Pie',

marker\_colors = clr

))

fig.update\_layout(title = {'text': "Volume of Casualties For Each Casualty Class",

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'})

fig.show()

# Merge casualty and accident dataframe on accident index

casualty\_new = casualty.merge(accident, on = 'Accident\_Index')

casualty\_new.head(10)

# Count of casualties for each age band

ab\_dict = dict(casualty\_new[casualty\_new['Age\_Band\_of\_Casualty']!=-1]['Age\_Band\_of\_Casualty'].value\_counts())

ab\_dict

ab\_df = pd.DataFrame(zip(ab\_dict.keys(), list(ab\_dict.values())),

columns = ['Age\_Band', 'Casualty\_Count'])

ab\_df

ab\_df = ab\_df.sort\_values(by=['Age\_Band'])

ab\_df

ab\_df['Age\_Band'] = ab\_df['Age\_Band'].replace(

to\_replace = [1,2,3,4,5,6,7,8,9,10,11], value = ['0 - 5','6 - 10','11 - 15','16 - 20','21 - 25','26 - 35','36 - 45',

'46 - 55','56 - 65','66 - 75','Over 75'])

ab\_df

fig = go.Figure(data=[go.Bar(

x=ab\_df['Age\_Band'],

y=ab\_df['Casualty\_Count'],

marker\_color='goldenrod'

)])

fig.update\_layout(title={'text':'Count of Casualties in Each Age Band',

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'},

xaxis\_title = 'Age Band of Casualty',

yaxis\_title = 'Count of Casualties')

# Dictionary for count of casualties for each gender

casualty\_sex\_dict = dict(casualty['Sex\_of\_Casualty'].value\_counts())

casualty\_sex\_dict

casualty\_sex\_df = pd.DataFrame(zip(casualty\_sex\_dict.keys(), list(casualty\_sex\_dict.values())),

columns = ['Sex\_of\_Casualty', 'Casualty\_Count'])

casualty\_sex\_df

casualty\_sex\_df['Sex\_of\_Casualty'] = casualty\_sex\_df['Sex\_of\_Casualty'].replace(

to\_replace = [1,2,-1], value = ['Male','Female','Data missing or out of range'])

casualty\_sex\_df

clr = ['#6A5ACD','#FF82AB','#848484']

fig = go.Figure()

fig.add\_trace(go.Pie(

labels=casualty\_sex\_df['Sex\_of\_Casualty'],

values=casualty\_sex\_df['Casualty\_Count'],

textinfo='label+percent',

name='Casualty\_Sex\_Pie',

marker\_colors = clr

))

fig.update\_layout(title = {'text': "Volume of Casualties For Each Casualty Gender",

'y':0.9,

'x':0.5,

'xanchor': 'center',

'yanchor': 'top'})

pred\_accident = accident.drop(['Date','Did\_Police\_Officer\_Attend\_Scene\_of\_Accident','Hour'],

axis = 1).reset\_index(drop = True)

pred\_accident.head(10)

pred\_accident['Accident\_Index'] = pred\_accident['Accident\_Index'].astype(str)

# Calculation for majority gender of casualties

casualty\_gender\_count = casualty.groupby(['Accident\_Index', 'Sex\_of\_Casualty']).aggregate({'Sex\_of\_Casualty':sum})

casualty\_gender\_count[:20]

casualty\_gender\_count.columns = ['Sum']

casualty\_gender\_count = pd.DataFrame(casualty\_gender\_count).reset\_index()

casualty\_gender\_count['Accident\_Index'] = casualty\_gender\_count['Accident\_Index'].astype(str)

casualty['Accident\_Index'] = casualty['Accident\_Index'].astype(str)

accident['Accident\_Index'] = accident['Accident\_Index'].astype(str)

list\_accidentIndex = []

for x in accident['Accident\_Index']:

    list\_accidentIndex.append(x)

list\_accidentIndex[:10]

count\_male = {}

count\_female = {}

for item in list\_accidentIndex:

    count\_male[item] = casualty[(casualty['Accident\_Index'] == item) & (casualty['Sex\_of\_Casualty'] == 1)]['Accident\_Index'].count()

    count\_female[item] = casualty[(casualty['Accident\_Index'] == item) & casualty['Sex\_of\_Casualty'] == 2)]['Accident\_Index'].count()

import csv

csv\_columns1 = ['Accident\_Index','Count\_Male']

csv\_columns2 = ['Accident\_Index','Count\_Female']

csv\_file1 = "GenderCount\_Male\_new.csv"

csv\_file2 = "GenderCount\_Female\_new.csv"

with open(csv\_file1, 'w') as a\_file:

  writer = csv.writer(a\_file)

  for key, value in count\_male.items():

      writer.writerow([key, value])

with open(csv\_file2, 'w') as b\_file:

  writer = csv.writer(b\_file)

  for key, value in count\_female.items():

      writer.writerow([key, value])

#Add general nature of the sex of casualties for an accident

casualty\_gender = pd.read\_csv('Accident\_Casualty\_Gender.csv', sep='\t')

casualty\_gender['Accident\_Index'] = casualty\_gender['Accident\_Index'].astype(str)

pred\_accident = pd.merge(pred\_accident, casualty\_gender, on = 'Accident\_Index')

casualty\_gender.head(10)

for index in casualty\_gender.index:

  if casualty\_gender['Diff\_Male\_Female'][index] > 0:

    casualty\_gender['Casualty\_Gender'][index] = 'Mostly Male'

  elif casualty\_gender['Diff\_Male\_Female'][index] < 0:

    casualty\_gender['Casualty\_Gender'][index] = 'Mostly Female'

  else:

    casualty\_gender['Casualty\_Gender'][index] = 'Both gender'

casualty\_gender['Accident\_Index'] = list\_accidentIndex

casualty\_gender.head(10)

casualty\_gender.to\_csv('Accident\_Casualty\_Gender\_final.csv', index = False)

# Calculation for average age of casualties

avg\_age = {}

for index in list\_accidentIndex[400000:]:

  avg\_age[index] = st.mean(casualty[casualty['Accident\_Index'] == index]['Age\_of\_Casualty'])

csv\_file = "Casualty\_Age.csv"

with open(csv\_file, 'w') as a\_file:

  writer = csv.writer(a\_file)

  for key, value in avg\_age.items():

      writer.writerow([key, value])

casualty\_age = pd.read\_csv('Casualty\_Age.csv')

casualty\_age['Accident\_Index'] = list\_accidentIndex

casualty\_age['Accident\_Index'] = casualty\_age['Accident\_Index'].astype(str)

pred\_accident = pd.merge(pred\_accident, casualty\_age, on = 'Accident\_Index')

pred\_accident = pred\_accident.drop(['Accident\_Index','Count\_Male','Count\_Female','Diff\_Male\_Female'], axis = 1).reset\_index(drop = True)

pred\_accident.head(10)

pred\_accident[['Hour', 'Min']] = pred\_accident['Time'].str.split(':', expand = True)

pred\_accident = pred\_accident.drop(['Time'], axis = 1)

pred\_accident.head(10)

pred\_accident.to\_csv('pred\_accident.csv', index = False)

# Prediction for accident severity

import statistics as st

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import cross\_val\_score

from sklearn import tree

from imblearn.over\_sampling import SMOTE

from imblearn.under\_sampling import RandomUnderSampler

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn import metrics

from collections import Counter

from sklearn.metrics import classification\_report, confusion\_matrix

pred\_accident = pd.read\_csv('/content/drive/My Drive/AIT/Thesis/pred\_accident.csv')

pred\_accident\_severity = pred\_accident.drop(['Number\_of\_Vehicles','Number\_of\_Casualties', 'Local\_Authority\_(Highway)', 'Casualty\_Gender', 'Avg\_Casualty\_Age'], axis = 1).reset\_index(drop = True)

sns.set\_style('darkgrid')

sns.set\_palette('bright')

sns.countplot(x="Accident\_Severity", data=pred\_accident\_severity)

pred\_accident\_severity['Accident\_Severity'].value\_counts()

# Proportion of accident severities

sns.set\_palette('muted')

pred\_accident['Accident\_Severity'].value\_counts().plot(kind = 'pie', autopct='%1.1f%%', startangle=90, figsize=(5, 5),

                                              title = 'Accident Severity')

plt.legend(['3 - Slight', '2 - Serious', '1 - Fatal'], loc = 'best')

# Original Data

X = pred\_accident\_severity.drop(['Accident\_Severity'], axis = 1)

y = pred\_accident\_severity['Accident\_Severity']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

Counter(y\_train)

Counter(y\_test)

# Decision Tree model

model\_DTree = DecisionTreeClassifier()

model\_DTree = model\_DTree.fit(X\_train, y\_train)

y\_pred = model\_DTree.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Random Forest model

rfc = RandomForestClassifier()

rfc.fit(X\_train,y\_train)

y\_pred = rfc.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Naive Bayes model

gnb = GaussianNB().fit(X\_train, y\_train)

gnb\_predictions = gnb.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Minority Class Oversampling using SMOTE

# Oversampling the minority class in the data using SMOTE

sm = SMOTE()

X\_over, y\_over = sm.fit\_resample(X, y)

X\_train\_over, X\_test\_over, y\_train\_over, y\_test\_over = train\_test\_split(X\_over, y\_over, test\_size=0.3, random\_state=42)

Counter(y\_train\_over)

Counter(y\_test\_over)

# Decision Tree model

model\_DTree\_over = DecisionTreeClassifier()

model\_DTree\_over = model\_DTree\_over.fit(X\_train\_over, y\_train\_over)

y\_pred\_over = model\_DTree\_over.predict(X\_test\_over)

print("Accuracy:",metrics.accuracy\_score(y\_test\_over, y\_pred\_over))

print(confusion\_matrix(y\_test\_over, y\_pred\_over))

print(classification\_report(y\_test\_over, y\_pred\_over))

# Random Forest model

rfc\_over = RandomForestClassifier()

rfc\_over.fit(X\_train\_over,y\_train\_over)

y\_pred\_over = rfc\_over.predict(X\_test\_over)

print("Accuracy:",metrics.accuracy\_score(y\_test\_over, y\_pred\_over))

print(confusion\_matrix(y\_test\_over, y\_pred\_over))

print(classification\_report(y\_test\_over, y\_pred\_over))

# Naive Bayes model

gnb\_model\_over = GaussianNB().fit(X\_train\_over, y\_train\_over)

y\_pred\_over = gnb\_model\_over.predict(X\_test\_over)

print("Accuracy:",metrics.accuracy\_score(y\_test\_over, y\_pred\_over))

print(confusion\_matrix(y\_test\_over, y\_pred\_over))

print(classification\_report(y\_test\_over, y\_pred\_over))

# Majority Class Undersampling using RandomUnderSampler

# Undersampling the majority class in the data using RandomUnderSampler

rus = RandomUnderSampler(random\_state=42)

X\_under, y\_under = rus.fit\_resample(X, y)

X\_train\_under, X\_test\_under, y\_train\_under, y\_test\_under = train\_test\_split(X\_under, y\_under, test\_size=0.3, random\_state=42)

Counter(y\_train\_under)

Counter(y\_test\_under)

#Decision Tree model

model\_DTree\_under = DecisionTreeClassifier()

model\_DTree\_under = model\_DTree\_under.fit(X\_train\_under, y\_train\_under)

y\_pred\_under = model\_DTree\_under.predict(X\_test\_under)

print("Accuracy:",metrics.accuracy\_score(y\_test\_under, y\_pred\_under))

print(confusion\_matrix(y\_test\_under, y\_pred\_under))

print(classification\_report(y\_test\_under, y\_pred\_under))

# Random Forest model

from sklearn.ensemble import RandomForestClassifier

rfc\_under = RandomForestClassifier()

rfc\_under.fit(X\_train\_under,y\_train\_under)

y\_pred\_under = rfc\_under.predict(X\_test\_under)

print("Accuracy:",metrics.accuracy\_score(y\_test\_under, y\_pred\_under))

print(confusion\_matrix(y\_test\_under, y\_pred\_under))

print(classification\_report(y\_test\_under, y\_pred\_under))

# Naive Bayes model

gnb\_model\_under = GaussianNB().fit(X\_train\_under, y\_train\_under)

y\_pred\_under = gnb\_model\_under.predict(X\_test\_under)

print("Accuracy:",metrics.accuracy\_score(y\_test\_under, y\_pred\_under))

print(confusion\_matrix(y\_test\_under, y\_pred\_under))

print(classification\_report(y\_test\_under, y\_pred\_under))

# Prediction for number of casualties

from sklearn import metrics

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, confusion\_matrix

from sklearn.metrics import mean\_squared\_error, r2\_score

from collections import Counter

from scipy import stats

from scipy.stats import skew

from scipy.special import boxcox,inv\_boxcox

from sklearn.linear\_model import LinearRegression

import statsmodels.regression.linear\_model as sm

from sklearn.ensemble import RandomForestRegressor

from sklearn.neighbors import KNeighborsRegressor

from sklearn.model\_selection import GridSearchCV

from statsmodels.api import qqplot

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor as vif

pred\_accident\_cc = pred\_accident.drop(['Number\_of\_Vehicles', 'Local\_Authority\_(Highway)', 'Casualty\_Gender', 'Avg\_Casualty\_Age'],  axis = 1).reset\_index(drop = True)

pred\_accident\_cc.head(5)

sns.set\_style('whitegrid')

sns.set\_context("paper")

sns.distplot(pred\_accident\_cc['Number\_of\_Casualties'])

fig = go.Figure(data = [go.Histogram(x=pred\_accident\_cc['Number\_of\_Casualties'])])

fig.show()

skew(pred\_accident\_cc['Number\_of\_Casualties'])

pred\_accident\_cc['Number\_of\_Casualties'].describe()

Counter(pred\_accident\_cc['Number\_of\_Casualties'])

# Function to calculate VIF

def calc\_vif(X):

    vif\_df = pd.DataFrame()

    vif\_df["variables"] = X.columns

    vif\_df["VIF"] = [vif(X.values, i) for i in range(X.shape[1])]

    return(vif\_df)

X = pred\_accident\_cc.iloc[:,:-1]

calc\_vif(X)

pred\_accident\_cc\_dummy = pred\_accident\_cc.drop(['Location\_Easting\_OSGR','Location\_Northing\_OSGR','Longitude','Latitude','Local\_Authority\_(District)'],  axis = 1)

pred\_accident\_cc\_dummy.head(5)

X = pred\_accident\_cc\_dummy.iloc[:,:-1]

calc\_vif(X)

#Conversion of label encoded data to dummy encoded data

pred\_accident\_cc\_dummy = pd.get\_dummies(pred\_accident\_cc\_dummy, columns = ['Police\_Force', 'Accident\_Severity', 'Day\_of\_Week', '1st\_Road\_Class', 'Road\_Type', 'Junction\_Detail', 'Junction\_Control', '2nd\_Road\_Class', 'Pedestrian\_Crossing-Human\_Control', 'Pedestrian\_Crossing-Physical\_Facilities', 'Light\_Conditions', 'Weather\_Conditions', 'Road\_Surface\_Conditions', 'Special\_Conditions\_at\_Site', 'Carriageway\_Hazards', 'Urban\_or\_Rural\_Area'])

pred\_accident\_cc\_dummy.head(5)

X = pred\_accident\_cc\_dummy.drop(['Number\_of\_Casualties'], axis = 1)

y = pred\_accident\_cc\_dummy['Number\_of\_Casualties']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Linear regression using OLS method

reg\_OLS = sm.OLS(endog = y\_train, exog = X\_train).fit()

reg\_OLS.summary()

y\_pred = reg\_OLS.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': np.round(y\_pred,0)})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

qqplot(reg\_OLS.resid)

sns.set\_style('whitegrid')

sns.set\_context("paper")

sns.distplot(reg\_OLS.resid)

Counter(np.round(y\_pred,0))

plt.scatter(x=reg\_OLS.fittedvalues,y=reg\_OLS.resid,edgecolor='k')

plt.xlabel("Fitted values",fontsize=15)

plt.ylabel("Residuals",fontsize=15)

plt.title("Fitted vs. residuals plot",fontsize=18)

plt.grid(True)

sns.distplot(pred\_accident\_cc['Number\_of\_Casualties'])

Counter(pred\_accident\_cc['Number\_of\_Casualties'])

pred\_accident\_rf = pred\_accident\_cc.copy()

pred\_accident\_rf.head(5)

pred\_accident\_rf = pred\_accident\_rf.drop(['Location\_Easting\_OSGR','Location\_Northing\_OSGR','Longitude','Latitude','Local\_Authority\_(District)'],  axis = 1)

pred\_accident\_rf.head(5)

X = pred\_accident\_rf.drop(['Number\_of\_Casualties'], axis = 1)

y = pred\_accident\_rf['Number\_of\_Casualties']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Random Forest Regressor

reg\_rf = RandomForestRegressor()

reg\_rf.fit(X\_train, y\_train)

y\_pred = reg\_rf.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': np.round(y\_pred,0)})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

# KNN Regressor

rmse\_val = []

k = [50, 100, 150, 200, 250, 300]

for K in k:

    model = KNeighborsRegressor(n\_neighbors = K)

    model.fit(X\_train, y\_train)  #fit the model

    pred=model.predict(X\_test) #make prediction on test set

    error = np.sqrt(mean\_squared\_error(y\_test,pred)) #calculate rmse

    rmse\_val.append(error) #store rmse values

    print('RMSE value for k= ' , K , 'is:', error)

rmse\_val = [] #to store rmse values for different k

k = [350, 400, 450, 500, 550, 600]

for K in k:

    model = KNeighborsRegressor(n\_neighbors = K)

    model.fit(X\_train, y\_train)  #fit the model

    pred=model.predict(X\_test) #make prediction on test set

    error = np.sqrt(mean\_squared\_error(y\_test,pred)) #calculate rmse

    rmse\_val.append(error) #store rmse values

    print('RMSE value for k= ' , K , 'is:', error)

# k = 450 is selected since rmse is lowest

reg\_knn = KNeighborsRegressor(n\_neighbors = 450)

reg\_knn.fit(X\_train, y\_train)

y\_pred = reg\_knn.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': np.round(y\_pred,0)})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

#Prediction for number of vehicles

pred\_accident\_vc = pred\_accident.drop(['Number\_of\_Casualties', 'Local\_Authority\_(Highway)', 'Casualty\_Gender', 'Avg\_Casualty\_Age'], axis = 1).reset\_index(drop = True)

pred\_accident\_vc.head(5)

pred\_accident\_vc2 = pred\_accident\_vc.copy()

pred\_accident\_vc2['Number\_of\_Vehicles'].describe()

sns.set\_style('whitegrid')

sns.set\_context("paper")

sns.distplot(pred\_accident\_vc2['Number\_of\_Vehicles'])

fig = go.Figure(data = [go.Histogram(x=pred\_accident\_vc2['Number\_of\_Vehicles'])])

fig.show()

Counter(pred\_accident\_vc2['Number\_of\_Vehicles'])

X = pred\_accident\_vc2.iloc[:,:-1]

calc\_vif(X)

pred\_accident\_vc2 = pred\_accident\_vc2.drop(['Location\_Easting\_OSGR','Location\_Northing\_OSGR','Longitude','Latitude','Local\_Authority\_(District)'], axis = 1)

pred\_accident\_vc3 = pd.get\_dummies(pred\_accident\_vc2, columns= ['Police\_Force', 'Accident\_Severity', 'Day\_of\_Week','1st\_Road\_Class', 'Road\_Type', 'Junction\_Detail','Junction\_Control', '2nd\_Road\_Class', 'Pedestrian\_Crossing-Human\_Control', 'Pedestrian\_Crossing-Physical\_Facilities', 'Light\_Conditions', 'Weather\_Conditions', 'Road\_Surface\_Conditions', 'Special\_Conditions\_at\_Site', 'Carriageway\_Hazards', 'Urban\_or\_Rural\_Area'])

X = pred\_accident\_vc3.drop(['Number\_of\_Vehicles'], axis = 1)

y = pred\_accident\_vc3['Number\_of\_Vehicles']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Linear regression using OLS method

reg\_OLS = sm.OLS(endog = y\_train, exog = X\_train).fit()

reg\_OLS.summary()

y\_pred = reg\_OLS.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': np.round(y\_pred,0)})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

qqplot(reg\_OLS.resid)

sns.distplot(reg\_OLS.resid)

Counter(np.round(y\_pred,0))

plt.scatter(x=reg\_OLS.fittedvalues,y=reg\_OLS.resid,edgecolor='k')

plt.xlabel("Fitted values",fontsize=15)

plt.ylabel("Residuals",fontsize=15)

plt.title("Fitted vs. residuals plot",fontsize=18)

plt.grid(True)

pred\_accident\_vc2.head(5)

X = pred\_accident\_vc2.drop(['Number\_of\_Vehicles'], axis = 1)

y = pred\_accident\_vc2['Number\_of\_Vehicles']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Random Forest Regressor

reg\_rf = RandomForestRegressor()

reg\_rf.fit(X\_train, y\_train)

y\_pred = reg\_rf.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': np.round(y\_pred,0)})

pred\_df.head(10)

Counter(np.round(y\_pred,0))

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

# KNN Regressor

rmse\_val = []

k = [50, 100, 150, 200, 250, 300]

for K in k:

    model = KNeighborsRegressor(n\_neighbors = K)

    model.fit(X\_train, y\_train)  #fit the model

    pred=model.predict(X\_test) #make prediction on test set

    error = np.sqrt(mean\_squared\_error(y\_test,pred)) #calculate rmse

    rmse\_val.append(error) #store rmse values

    print('RMSE value for k= ' , K , 'is:', error)

# k = 150 is selected since rmse is lowest

reg\_knn = KNeighborsRegressor(n\_neighbors = 150)

reg\_knn.fit(X\_train, y\_train)

y\_pred = reg\_knn.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': np.round(y\_pred,0)})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

# Prediction for gender of casualties

# Dataframe for casualty gender prediction

pred\_accident\_gender = pred\_accident.drop(['Number\_of\_Vehicles', 'Local\_Authority\_(Highway)', 'Avg\_Casualty\_Age'],  axis = 1).reset\_index(drop = True)

pred\_accident\_gender.head(5)

pred\_accident\_gender['Casualty\_Gender'].value\_counts()

sns.set\_style('darkgrid')

sns.set\_palette('bright')

sns.countplot(x="Casualty\_Gender", data=pred\_accident\_gender)

#Proportion of each casualty gender

sns.set\_palette('muted')

pred\_accident\_gender['Casualty\_Gender'].value\_counts().plot(kind = 'pie', autopct='%1.1f%%', startangle=90, figsize=(5, 5), title = 'Casualty Gender')

# Original Data

X = pred\_accident\_gender.drop(['Casualty\_Gender'], axis = 1)

y = pred\_accident\_gender['Casualty\_Gender']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

Counter(y\_train)

Counter(y\_test)

# Decision Tree model

model\_DTree = DecisionTreeClassifier()

model\_DTree = model\_DTree.fit(X\_train, y\_train)

y\_pred = model\_DTree.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Random Forest model

rfc\_model = RandomForestClassifier()

rfc\_model.fit(X\_train,y\_train)

y\_pred = rfc\_model.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Naive Bayes model

gnb = GaussianNB()

gnb = gnb.fit(X\_train, y\_train)

y\_pred = gnb.predict(X\_test)

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

# Minority Class Oversampling using SMOTE

# Oversampling the minority class in the data using SMOTE

sm = SMOTE()

X\_over, y\_over = sm.fit\_resample(X, y)

X\_train\_over, X\_test\_over, y\_train\_over, y\_test\_over = train\_test\_split(X\_over, y\_over, test\_size=0.3, random\_state=42)

Counter(y\_train\_over)

Counter(y\_test\_over)

# Decision Tree model

model\_DTree\_over = DecisionTreeClassifier()

model\_DTree\_over.fit(X\_train\_over, y\_train\_over)

y\_pred\_over = model\_DTree\_over.predict(X\_test\_over)

print("Accuracy:",metrics.accuracy\_score(y\_test\_over, y\_pred\_over))

print(confusion\_matrix(y\_test\_over, y\_pred\_over))

print(classification\_report(y\_test\_over, y\_pred\_over))

# Random Forest model

rfc\_over = RandomForestClassifier()

rfc\_over.fit(X\_train\_over,y\_train\_over)

y\_pred\_over = rfc\_over.predict(X\_test\_over)

print("Accuracy:",metrics.accuracy\_score(y\_test\_over, y\_pred\_over))

print(confusion\_matrix(y\_test\_over, y\_pred\_over))

print(classification\_report(y\_test\_over, y\_pred\_over))

# Naive Bayes model

gnb\_model\_over = GaussianNB()

gnb\_model\_over.fit(X\_train\_over, y\_train\_over)

y\_pred\_over = gnb\_model\_over.predict(X\_test\_over)

print("Accuracy:",metrics.accuracy\_score(y\_test\_over, y\_pred\_over))

print(confusion\_matrix(y\_test\_over, y\_pred\_over))

print(classification\_report(y\_test\_over, y\_pred\_over))

# Majority Class Undersampling using RandomUnderSampler

# Undersampling the majority class in the data using RandomUnderSampler

rus = RandomUnderSampler(random\_state=42)

X\_under, y\_under = rus.fit\_resample(X, y)

X\_train\_under, X\_test\_under, y\_train\_under, y\_test\_under = train\_test\_split(X\_under, y\_under, test\_size=0.3, random\_state=42)

Counter(y\_train\_under)

Counter(y\_test\_under)

#Decision Tree model

model\_DTree\_under = DecisionTreeClassifier()

model\_DTree\_under.fit(X\_train\_under, y\_train\_under)

y\_pred\_under = model\_DTree\_under.predict(X\_test\_under)

print("Accuracy:",metrics.accuracy\_score(y\_test\_under, y\_pred\_under))

print(confusion\_matrix(y\_test\_under, y\_pred\_under))

print(classification\_report(y\_test\_under, y\_pred\_under))

# Random Forest model

rfc\_under = RandomForestClassifier()

rfc\_under.fit(X\_train\_under,y\_train\_under)

y\_pred\_under = rfc\_under.predict(X\_test\_under)

print("Accuracy:",metrics.accuracy\_score(y\_test\_under, y\_pred\_under))

print(confusion\_matrix(y\_test\_under, y\_pred\_under))

print(classification\_report(y\_test\_under, y\_pred\_under))

#Naive Bayes model

gnb\_model\_under = GaussianNB()

gnb\_model\_under.fit(X\_train\_under, y\_train\_under)

y\_pred\_under = gnb\_model\_under.predict(X\_test\_under)

print("Accuracy:",metrics.accuracy\_score(y\_test\_under, y\_pred\_under))

print(confusion\_matrix(y\_test\_under, y\_pred\_under))

print(classification\_report(y\_test\_under, y\_pred\_under))

# Prediction for average age of casualties

pred\_accident\_age = pred\_accident.drop(['Local\_Authority\_(Highway)'], axis = 1).reset\_index(drop = True)

pred\_accident\_age.head(5)

sns.set\_style('whitegrid')

sns.set\_context("paper")

sns.distplot(pred\_accident\_age['Avg\_Casualty\_Age'])

pred\_accident\_age['Avg\_Casualty\_Age'].describe()

# Count and remove unknown data from Avg\_Casualty\_Age column

pred\_accident\_age[pred\_accident\_age['Avg\_Casualty\_Age'] == -1].count()

pred\_accident\_age = pred\_accident\_age[pred\_accident\_age['Avg\_Casualty\_Age'] > 0]

pred\_accident\_age['Avg\_Casualty\_Age'].describe()

sns.set\_style('whitegrid')

sns.set\_context("paper")

sns.distplot(pred\_accident\_age['Avg\_Casualty\_Age'])

pred\_accident\_age2 = pd.get\_dummies(pred\_accident\_age, columns= ['Police\_Force', 'Accident\_Severity', 'Day\_of\_Week','1st\_Road\_Class', 'Road\_Type', 'Junction\_Detail', 'Junction\_Control', '2nd\_Road\_Class', 'Pedestrian\_Crossing-Human\_Control', 'Pedestrian\_Crossing-Physical\_Facilities', 'Light\_Conditions', 'Weather\_Conditions', 'Road\_Surface\_Conditions', 'Special\_Conditions\_at\_Site', 'Carriageway\_Hazards', 'Urban\_or\_Rural\_Area', 'Casualty\_Gender'])

pred\_accident\_age2.head(5)

X = pred\_accident\_age2.drop(['Avg\_Casualty\_Age'], axis = 1)

y = pred\_accident\_age2['Avg\_Casualty\_Age']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

X = pred\_age\_vif.iloc[:,:-1]

calc\_vif(X)

# Police force indicates the area, so coordinates and district name are not required

pred\_age\_ols = pred\_accident\_age.drop(['Location\_Easting\_OSGR','Location\_Northing\_OSGR','Longitude','Latitude','Local\_Authority\_(District)'],  axis = 1)

pred\_age\_ols = pd.get\_dummies(pred\_age\_ols, columns = ['Casualty\_Gender'])

pred\_age\_ols.head(5)

X = pred\_age\_ols.iloc[:,:-1]

calc\_vif(X)

X = pred\_age\_ols.drop(['Avg\_Casualty\_Age'], axis = 1)

y = pred\_age\_ols['Avg\_Casualty\_Age']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

reg\_OLS = sm.OLS(endog = y\_train, exog = X\_train).fit()

reg\_OLS.summary()

y\_pred = reg\_OLS.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

qqplot(reg\_OLS.resid)

sns.distplot(reg\_OLS.resid)

plt.scatter(x=reg\_OLS.fittedvalues,y=reg\_OLS.resid,edgecolor='k')

plt.xlabel("Fitted values",fontsize=15)

plt.ylabel("Residuals",fontsize=15)

plt.title("Fitted vs. residuals plot",fontsize=18)

plt.grid(True)

pred\_age\_rf = pred\_age\_ols.copy()

pred\_age\_rf.head(5)

X = pred\_age\_rf.drop(['Avg\_Casualty\_Age'], axis = 1)

y = pred\_age\_rf['Avg\_Casualty\_Age']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Random Forest Regressor

reg\_rf = RandomForestRegressor()

reg\_rf.fit(X\_train, y\_train)

y\_pred = reg\_rf.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))

# KNN Regressor

rmse\_val = []

k = [50, 100, 150, 200, 250, 300]

for K in k:

    model = KNeighborsRegressor(n\_neighbors = K)

    model.fit(X\_train, y\_train)  #fit the model

    pred=model.predict(X\_test) #make prediction on test set

   error = np.sqrt(mean\_squared\_error(y\_test,pred)) #calculate rmse

    rmse\_val.append(error) #store rmse values

    print('RMSE value for k= ' , K , 'is:', error)

# k = 150 is selected since rmse is lowest

reg\_knn = KNeighborsRegressor(n\_neighbors = 150)

reg\_knn.fit(X\_train, y\_train)

y\_pred = reg\_knn.predict(X\_test)

pred\_df = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

pred\_df.head(10)

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

print('R2 Score:', r2\_score(y\_test, y\_pred))