University of Alberta

Navigating Road Risks: Trends and Predictions in Accident Data

Mohammad Shahriar Hossain
&
Sheikh Safwan Hossain

OM 420: Predictive Business Analytics

Maryam H. Mofard

April 14, 2024

Introduction

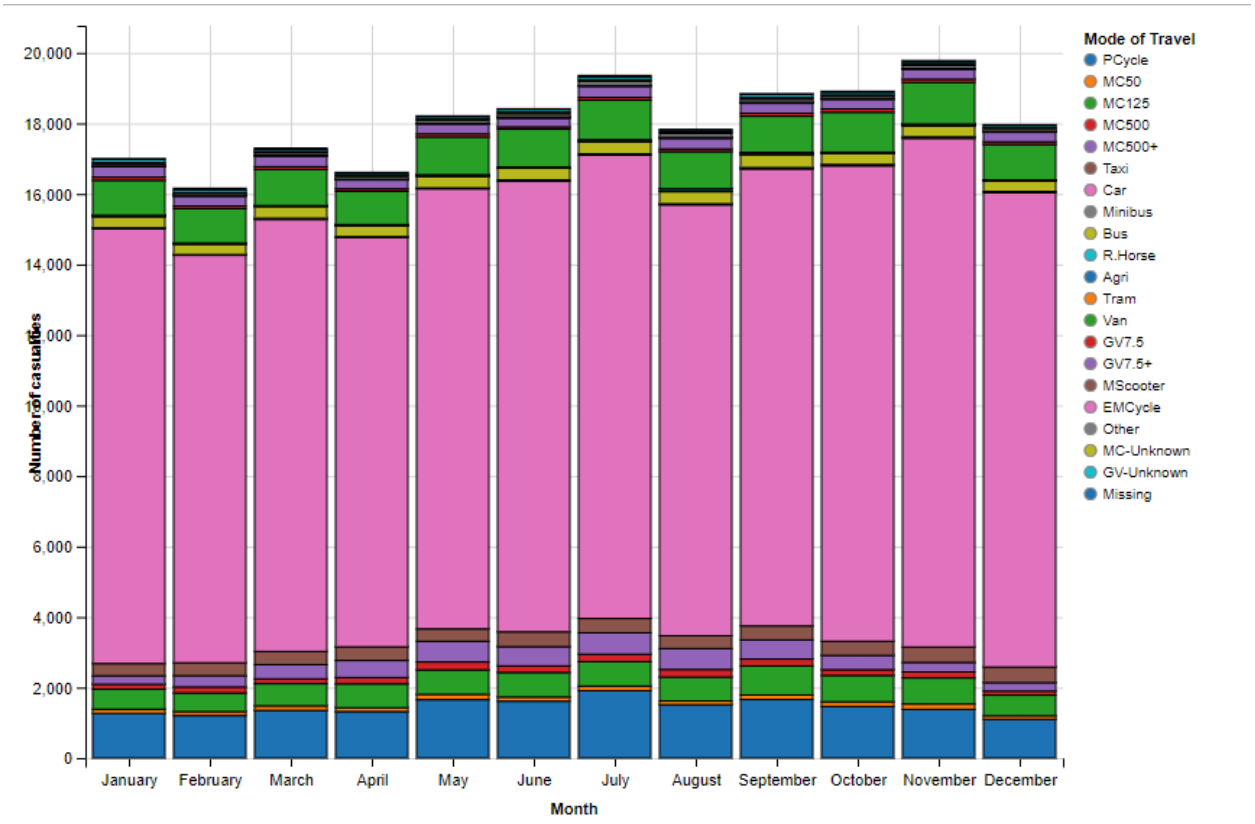
In reviewing 2019's road accident data, we aimed to discern patterns that could improve safety on public roads. Our analysis focused on the factors contributing to accident severity, with the goal to accurately predict such outcomes. The exploration revealed key insights, laying the groundwork for testing various classification models. Ultimately, the project seeks to identify the best model for predicting accident severity to aid in enhancing road safety measures.

Dataset

Our project uses three datasets from 2019's UK road accident records: Accidents, Vehicles, and Casualties. The Accidents dataset captures where, when, and how severe each incident was, while the Vehicles dataset looks at the types and conditions of vehicles involved. The Casualties dataset sheds light on who was affected and how seriously. All these details are connected through a unique identifier, the 'Accident_Index', letting us piece together a full picture of each accident to help enhance road safety.

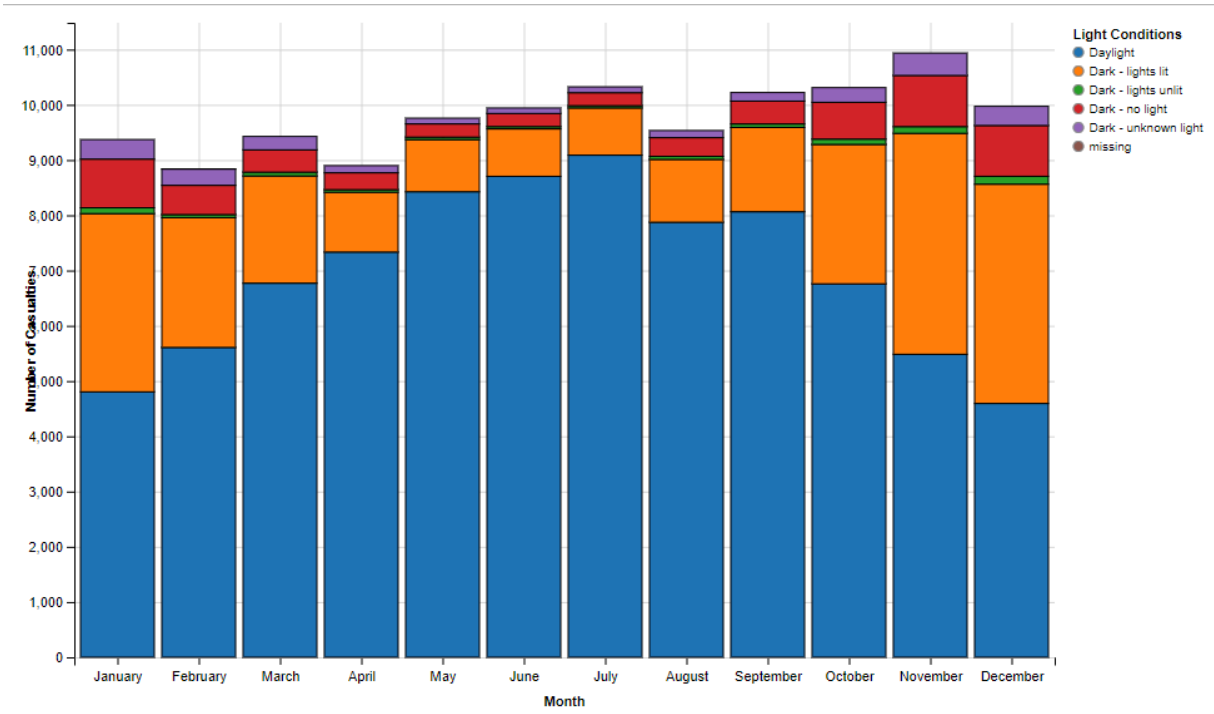
Observation 1 - Vehicle Type

When analyzing the types of vehicles involved in road accidents, it is apparent that cars lead significantly in accident involvement. The predominance of cars in accident statistics is illustrated in the provided graph. This is likely due to the fact that cars are the most common mode of transport, accounting for a larger share of road usage compared to other vehicle types. Consequently, their higher presence on the roads increases the probability of their involvement in accidents. It is important to consider these figures when developing traffic safety measures and interventions tailored to reduce car accidents.



Observation 2 - Light Condition

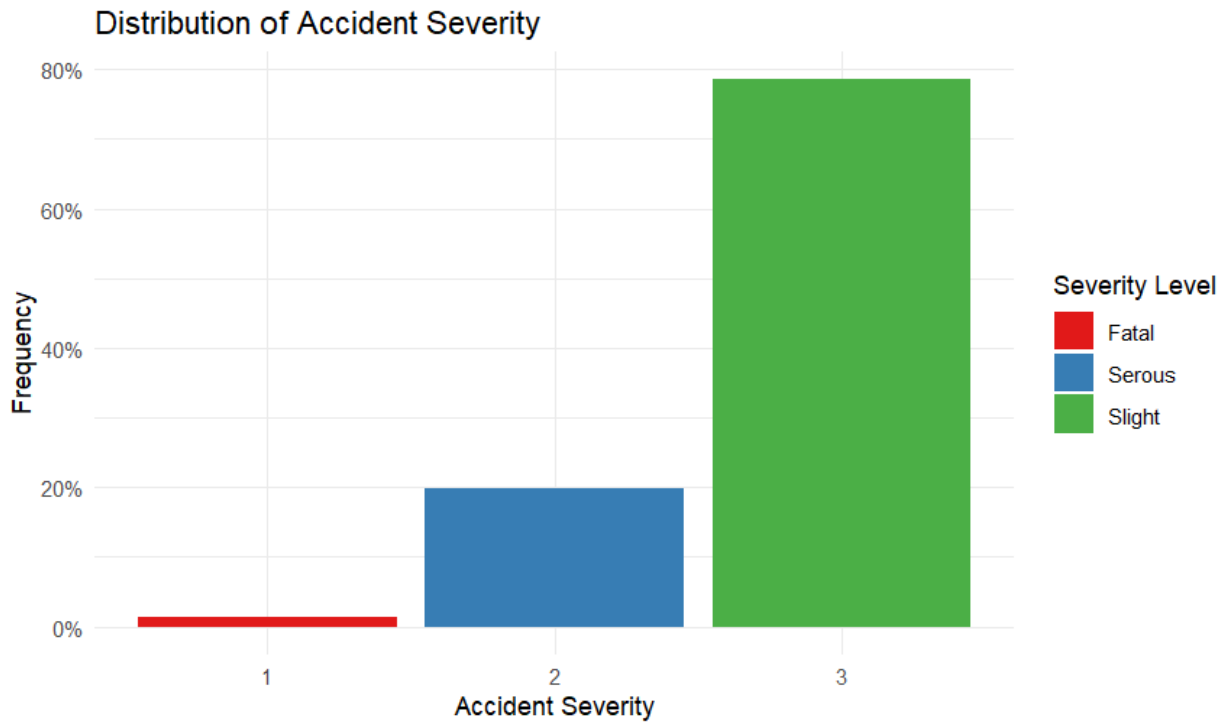
The distribution of accidents by light conditions reveals a consistent pattern across all months, with the majority of accidents occurring in daylight. This might initially seem counterintuitive, as daylight offers better visibility, but it can be explained by the increased volume of traffic during these hours. Additionally, during the summer months (May to August), the extended duration of daylight naturally coincides with an increase in daytime activities, which likely contributes to the higher number of accidents recorded in daylight during this period. This insight is valuable for planning safety campaigns that target the most active times of the day.



Observation 3 - Accident severity

Considering the severity of accidents, the data demonstrates a higher frequency of slight accidents compared to serious and fatal ones. This gradient in accident severity underscores the necessity for continuous efforts in road safety to not only prevent the most severe accidents but also to reduce the overall number of slight accidents, which constitute the bulk of incidents. Strategies to mitigate the factors leading to minor accidents could have a substantial impact on

improving overall traffic safety.



Data Preparation

To prepare the data for our study, we started by creating a binary target variable that classifies each accident's severity. Next, we simplified our datasets by removing unnecessary columns, ensuring that only relevant information was retained. Following this, we cleaned the data by removing any rows that contained missing or invalid entries. We then joined the accidents, vehicles, and casualties datasets by using left join on the accident index to match related records. After joining, we conducted another round of cleaning to eliminate any incomplete records that resulted from the merge. Key columns were then categorized appropriately for analysis by factoring the columns, and to ensure reproducibility, we established a random seed. Lastly, we divided the dataset into 70% training and 30% for validation and testing to support the various stages of developing and evaluating our predictive models.

Model 1: KNN

For the K-Nearest Neighbors (KNN) approach, two sets of variables were selected from the 18 predictors. These variable sets were employed to create two separate KNN models with (k = 5) neighbors. The first model utilized variables related to vehicle count, day of the week, road type, weather conditions, and casualty-related factors, among others. The second model used a different set of predictors including the number of casualties, speed limit, light conditions, and urban or rural settings. Error rates were calculated for both variable sets by comparing the predicted severity against the actual data.

KNN Result 1

		Predicted	
		1	2
Actual	1	1818	7190
	2	1893	27356

KNN Result 2

		Predicted	
		1	2
Actual	1	5451	3557
	2	418	28831

Model 2: LDA

In the Linear Discriminant Analysis (LDA) models, again, two variable sets were chosen from the 18 predictors. The first LDA model incorporated variables such as the number of vehicles, day of the week, and driver's age, while the second model considered factors like speed limit, light conditions, and casualty severity. Predictions were made against the test data, and error rates were evaluated to measure the performance of each variable set.

LDA Result 1

		Predicted	
		1	2
Actual	1	622	8386
	2	506	28743

LDA Result 2

		Predicted	
		1	2
Actual	1	6715	2293
	2	34	29215

Model 3: Decision Tree

The Decision Tree models were built using the same two sets of predictors as the LDA models. Parameters for the Decision Tree included a minimum split of 2, a minimum bucket size of 1, and a complexity parameter set to -1 with cross-validation (xval=5). The models aimed to classify the severity target variable based on the selected predictors. The performance of each tree was assessed, and the error rates for both sets of variables were computed to identify the most effective model.

CART Result 1

		Predicted	
		1	2
Actual	1	4128	4880
	2	4638	24611

CART Result 2

		Predicted	
		1	2
Actual	1	7942	1066
	2	920	28329

Conclusion

The performance of the various predictive models—K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and Random Forest—was assessed using confusion matrices,

which were instrumental in calculating the error rates for each model. When examining the error rates, it becomes evident that the LDA model exhibited the lowest error rate for the first set of variables, suggesting a strong performance in predicting accident severity. On the other hand, while the Random Forest model showed the best performance for the second set of variables, it had the highest error rate for the first set of variables. This inconsistency in the Random Forest model's performance across different variable sets led to the decision to favor the LDA model. The LDA's reliable performance across both sets of variables indicates a more robust and consistent model, making it the preferred choice for predicting the severity of road accidents.

	KNN	LDA	Random Forest
Error (variable set 1)	0.2374206	0.2324281	0.2487911
Error (variable set 2)	0.1039026	0.06082547	0.05191207

Appendix

```
rm(list = ls())
setwd("to\\the\\working\\directory\\where\\the\\files\\are")
accidents <- read.csv("Accidents 2019.csv")
vehicles <- read.csv("Vehicles 2019.csv")
casualties <- read.csv("Casualties 2019.csv")

# PART 1 (creating columns, factoring columns before EDA and EDA)
###############################################################################################
###############################################################################################

install.packages("ggvis")
library(ggvis)
library(dplyr)

##### Creating / converting necessary columns before Exploratory Data Analysis #####
###############################################################################################

# converting Date column from character to date format
accidents$Date <- as.Date(accidents$Date, "%d/%m/%Y")
class(accidents$Date)

# Creating a Month column in the accidents dataset
accidents$Month <- format(accidents$Date, format= "%B")
accidents$Month <- as.factor(accidents$Month)
accidents$Month <- factor(accidents$Month,levels = month.name)
summary(accidents$Month)

#Creating a Day column in the accidents dataset
accidents$Day <- factor(accidents$Day_of_Week,
                        levels = 1:7,
                        labels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"))
summary(accidents$Day)

# Extracting the hour part from the 'Time' column and creating an Hour column in the
accidents dataset
accidents$Hour <- as.integer(substr(accidents$Time, 1, 2))

library(dplyr)

# rewrite Accident_Severity into a new Severity column with descriptive names in the
accidents dataset
accidents <- accidents %>%
  mutate(Severity = case_when(
    Accident_Severity == 1 ~ "Fatal",
    Accident_Severity == 2 ~ "Serious",
    Accident_Severity == 3 ~ "Slight",
    TRUE ~ as.character(Accident_Severity)
  ))
```

```
# Creating new column Type in the vehicles dataset by shortening the Vehicle names
vehicles$Type <- factor(vehicles$Vehicle_Type,
                                levels =
c(1,2,3,4,5,8,9,10,11,16,17,18,19,20,21,22,23,90,97,98,-1),
                                labels = c("PCycle", "MC50", "MC125", "MC500", "MC500+",
                                           "Taxi", "Car", "Minibus", "Bus", "R.Horse",
                                           "Agri", "Tram", "Van", "GV7.5", "GV7.5+",
                                           "MScooter", "EMCycle", "Other", "MC-Unknown",
                                           "GV-Unknown", "Missing"))
levels(vehicles$Type)

# Creating new column Casualty_Sex in the casualties dataset for descriptive names
casualties$Casualty_Sex <- factor(casualties$Sex_of_Casualty,
                                  levels = c(1, 2, -1),
                                  labels = c("Male", "Female", "Missing"))

# Creating new column Light in the accidents dataset for descriptive names
accidents$Light <- factor(accidents$Light_Conditions,
                        levels = c(1,4,5,6,7,-1),
                        labels = c("Daylight", "Dark - lights lit", "Dark - lights unlit",
"Dark - no light", "Dark - unknown light", "missing"))

# Define age bands
bands <- c("0-4", "5-9", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39",
           "40-44", "45-49", "50-54", "55-59", "60-64", "65-69", "70-74", "75-79",
           "80-84", "85-89", "90-94", "95-99", "100-104", "105-109", "110-114", "115-120")

# Using cut() to create Age_Band in the casualties dataset
casualties$Age_Band <- cut(casualties$Age_of_Casualty,
                           breaks = c(0, 4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54, 59, 64,
69, 74, 79, 84, 89, 94, 99, 104, 109, 114, 120),
                           labels = bands,
                           right = FALSE)

##### Exploratory Data Analysis #####
################################################################################################

# Table: Counts of Vehicle Types

install.packages("knitr")
library(knitr)

vehicle_table <- table(vehicles$Type)

kable(vehicle_table,
      caption = "Counts of Vehicle Types",
      col.names = c("Vehicle Type", "Count"),
      align = "c")

# Table: Counts of Vehicle Types based on accident severity
library(dplyr)
```

```
library(tidyverse)

combined_data <- vehicles %>%
  select(Accident_Index, Type) %>%
  inner_join(accidents %>% select(Accident_Index, Severity), by = "Accident_Index")

table <- with(combined_data, table(Type, Severity))

table


# Distribution of Accident Severity
library(scales)

ggplot(accidents, aes(x = as.factor(Accident_Severity))) +
  geom_bar(aes(y = ..count../sum(..count..), fill = as.factor(Accident_Severity))) +  # Use
proportions
  scale_y_continuous(labels = percent_format()) +  # Convert y-axis to percentages
  labs(x = "Accident Severity", y = "Frequency", title = "Distribution of Accident
Severity") +
  scale_fill_brewer(palette = "Set1", labels = c("Fatal", "Serous", "Slight")) +
  theme_minimal() +
  guides(fill = guide_legend(title = "Severity Level"))  # Add a legend for severity level


# Casualties by mode of travel

library(ggvis)

vehicles %>%
  group_by(Type) %>%
  summarise(Number_of_Casualties = n()) %>%
  ggvis(x = ~Type, y = ~Number_of_Casualties, fill = ~factor(Type)) %>%
  layer_bars() %>%
  add_axis("x", title = "Mode of Travel") %>%
  add_axis("y", title = "Number of Vehicles Involved") %>%
  add_legend("fill", title = "Mode of Travel")


# Casualties by month
accidents %>%
  ggvis(~Month, fill :="#e5f5f9") %>%
  layer_bars() %>%
  add_axis("x", title = "Month") %>%
  add_axis("y", title = "Number of casualties")


# Casualties by month and mode of travel

combined_data <- accidents %>%
  left_join(vehicles, by = "Accident_Index")
```

```
combined_data <- combined_data %>%
  mutate(Month = format(as.Date(Date, format="%d/%m/%Y"), "%m")) %>%
  mutate(Month = factor(Month, levels = c("01", "02", "03", "04", "05", "06", "07", "08",
"09", "10", "11", "12"),
                        labels = c("January", "February", "March", "April", "May", "June",
                                   "July", "August", "September", "October", "November",
"December")))

combined_data %>%
  ggvis(x = ~Month, fill = ~as.factor(Type)) %>%
  layer_bars() %>%
  add_axis("x", title = "Month") %>%
  add_axis("y", title = "Number of casualties") %>%
  add_legend("fill", title = "Mode of Travel")


# casualties by age band and gender
casualties %>%
  group_by(Age_Band, Casualty_Sex) %>%
  summarise(Number_of_Casualties = n()) %>%
  ggvis(x = ~Age_Band, y = ~Number_of_Casualties, fill = ~Casualty_Sex) %>%
  layer_bars() %>%
  add_axis("x", title = "Age Band") %>%
  add_axis("y", title = "Number of Casualties") %>%
  add_legend("fill", title = "Casualty Sex")


# casualties by month and light conditions
accident_counts <- accidents %>%
  group_by(Month, Light) %>%
  summarise(Number_of_Casualties = n(), .groups = 'drop')  # Ensuring groups are dropped
after summarisation

accident_counts %>%
  ggvis(x = ~Month, y = ~Number_of_Casualties, fill = ~factor(Light)) %>%
  layer_bars() %>%
  add_axis("x", title = "Month") %>%
  add_axis("y", title = "Number of Casualties") %>%
  add_legend("fill", title = "Light Conditions")


# Boxplot for Mean age of casualties by mode of travel
combined_data <- left_join(vehicles, casualties, by = c("Accident_Index",
"Vehicle_Reference"))

ggplot(combined_data, aes(x = factor(Type), y = Age_of_Casualty)) +
  geom_boxplot() +
  xlab("Vehicle Type") +
  ylab("Age of Casualty") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) # Rotate x labels for readability
```

```
# line chart with ggvis for Casualties by day
combined_data <- accidents %>%
  left_join(casualties, by = "Accident_Index") %>%
  group_by(Date) %>%
  summarize(count = n(), .groups = 'drop') # count the number of casualties per date

combined_data %>%
  ggvis(~Date, ~count) %>%
  layer_lines() %>%
  add_axis("x", title = "Date") %>%
  add_axis("y", title = "Number of Casualties")


# Create a world map to find out from where these accident data are coming from

incomp_ids <- which(!complete.cases(accidents))

accidents_map <- na.omit(accidents)
any(!complete.cases(accidents_map))

install.packages("sf")
install.packages("rnaturalearth")
install.packages("rnaturalearthdata")

library(sf)
library(rnaturalearth)

world <- ne_countries(scale = "medium", returnclass = "sf")

accidents_sf <- st_as_sf(accidents_map, coords = c("Longitude", "Latitude"), crs = 4326)

ggplot(data = world) +
  geom_sf(fill = "lightskyblue3", color = "lightskyblue4") +
  geom_sf(data = accidents_sf, color = "black", size = 0.6, alpha = 0.8) +
  theme_minimal() +
  labs(x = "Longitude", y = "Latitude", title = "Accident Locations") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


# zoomed in UK map to see all the accidents (black dots)
install.packages("maps")
install.packages("mapdata")

library(maps)
library(mapdata)

ggplot(data = accidents_map) +
  borders("world", xlim = range(accidents_map$Longitude), ylim =
range(accidents_map$Latitude), fill = "steelblue3", colour = "navy") +
  geom_point(aes(x = Longitude, y = Latitude), color = "grey20", size = 0.5) +
  coord_fixed(ratio = 1) +
```

```
  labs(x = "Longitude", y = "Latitude", title = "Accident location zoomed (UK)") +
  scale_x_continuous(limits = range(accidents_map$Longitude)) +
  scale_y_continuous(limits = range(accidents_map$Latitude)) +
  theme_minimal()


# PART 2 (Cleaning and preparing the data before modelling, Building models (KNN, LDA and
Random Forest))
################################################################################################
################################################################################################

##### Cleaning and preparing the data #####
################################################################################################

accidents$Month <- match(accidents$Month, month.name)

# 1. create target variable (Severity_Target)
table(accidents$Accident_Severity)
accidents <- mutate(accidents, Severity_Target = ifelse(Accident_Severity %in% c(1, 2), 1,
2))
table(accidents$Severity_Target)

# 2. Delete unnecessary columns from the datasets
accidents = accidents[-c(2:7,10,12:16,19:24,28:29,31:32,34,36:37)]
vehicles = vehicles[-c(2,4:14,17:24)]
casualties = casualties[-c(2:3,7,9:18)]

# 3. removing rows where any column value is NA or -1
accidents_clean <- accidents %>%
  filter_all(all_vars(!is.na(.) & . != -1))

vehicles_clean <- vehicles %>%
  filter_all(all_vars(!is.na(.) & . != -1))

casualties_clean <- casualties %>%
  filter_all(all_vars(!is.na(.) & . != -1))

any(!complete.cases(accidents_clean))
any(!complete.cases(vehicles_clean))
any(!complete.cases(casualties_clean))

# 4. joining all 3 datasets
accidents_vehicles <- left_join(accidents_clean, vehicles_clean, by = "Accident_Index")
data <- left_join(accidents_vehicles, casualties_clean, by = "Accident_Index")

# 5. removing NA rows after joining
incomp_ids <- which(!complete.cases(data))
data_clean <- na.omit(data)
any(!complete.cases(data_clean))

# 6. factoring relevant columns
columns_to_factor <- c(4:5,7:15,17:18,20)
```

```
data_clean <- data_clean %>%
  mutate(across(all_of(columns_to_factor), as.factor))

sapply(data_clean, is.factor)

# write.csv(data_clean, "data.csv", row.names = FALSE)

# 7. set random seed
set.seed(123)

# 8. divide the dataset into three parts: train, test and validation
percen_train <- 0.7
percen_validation_test <- 0.3  # Remaining 30% for validation and test
percen_validation <- 0.5  # Half of the 30% will be 15% of the total data

train_indices <- sample(1:nrow(data_clean), percen_train * nrow(data_clean))
data_train <- data_clean[train_indices, ]

temp_data <- data_clean[-train_indices, ]

validation_indices <- sample(1:nrow(temp_data), percen_validation * nrow(temp_data))
data_validation <- temp_data[validation_indices, ]
data_test <- temp_data[-validation_indices, ]

##### Building models #####
#############################################################################################

##### KNN #####
#############################################################################################
library(class)
pred.train1 <- knn(data_train[,c(2,4,5,8,12,14,15,17,19)],
                   data_test[,c(2,4,5,8,12,14,15,17,19)],
                   data_train[,13], k=5)
knnResult1 <- table(data_test[,13],pred.train1)
knntest.error1<- 1-sum(diag(knnResult1))/sum(knnResult1)

pred.train2 <- knn(data_train[,c(3,6,7,9,10,11,16,18,20)],
                    data_test[,c(3,6,7,9,10,11,16,18,20)],
                    data_train[,13], k=5)
knnResult2 <- table(data_test[,13],pred.train2)
knntest.error2<- 1-sum(diag(knnResult2))/sum(knnResult2)

##### LDA #####
#############################################################################################
library(MASS)
lda.fit1 =
lda(Severity_Target~Number_of_Vehicles+Day_of_Week+Road_Type+Weather_Conditions+Hour+Vehicle
_Type+Sex_of_Driver+Casualty_Class+Age_of_Casualty, data=data_train)
lda.pred1 = predict(lda.fit1,data_test)
ldaResult1 <- table(data_test$Severity_Target, lda.pred1$class)
ldatest.error1<- 1-sum(diag(ldaResult1))/sum(ldaResult1)
```

```
lda.fit2 =
lda(Severity_Target~Number_of_Casualties+Speed_limit+Light_Conditions+Road_Surface_Condition
s+Urban_or_Rural_Area+Month+Age_of_Driver+Sex_of_Casualty+Casualty_Severity,
data=data_train)
lda.pred2 = predict(lda.fit2,data_test)
ldaResult2 <- table(data_test$Severity_Target, lda.pred2$class)
ldatest.error2 <- 1-sum(diag(ldaResult2))/sum(ldaResult2)

##### Decision Tree #####
################################################################################
install.packages("rpart")
library(rpart)
library(readr)
install.packages("rattle")
library(rattle)

fit1<-rpart(Severity_Target~Number_of_Vehicles+Day_of_Week+Road_Type+Weather_Conditions+Hour
+Vehicle_Type+Sex_of_Driver+Casualty_Class+Age_of_Casualty,
            data=data_train,method="class",parms=list(split="information
"), minsplit=2, minbucket=1, cp=-1,xval=5)
pred1 <- predict(fit1, data_test, type = "class")
cartResult1 <- table(data_test$Severity_Target, pred1)
carttest.error1<- 1-sum(diag(cartResult1))/sum(cartResult1)

fit2<-rpart(Severity_Target~Number_of_Casualties+Speed_limit+Light_Conditions+Road_Surface_C
onditions+Urban_or_Rural_Area+Month+Age_of_Driver+Sex_of_Casualty+Casualty_Severity,
             data=data_train,method="class",parms=list(split="information
"), minsplit=2, minbucket=1, cp=-1,xval=5)
pred2 <- predict(fit2, data_test, type = "class")
cartResult2 <- table(data_test$Severity_Target, pred2)
carttest.error2 <- 1-sum(diag(cartResult2))/sum(cartResult2)


# PART 3 (Visualizing the confusion matrices and error rate comparison))
################################################################################
################################################################################

# creating a plot confusion matrix function to show the matrices
library(ggplot2)
install.packages("reshape2")
library(reshape2)

plot_confusion_matrix <- function(cm, title) {
  cm_melted <- melt(cm)
  colnames(cm_melted) <- c("Actual", "Predicted", "Freq")

  ggplot(cm_melted, aes(x = Predicted, y = Actual, fill = Freq)) +
    geom_tile(color = "white") +
    scale_fill_gradient(low = "white", high = "steelblue") +
    geom_text(aes(label = Freq), vjust = 1) +
    labs(title = title, x = 'Predicted', y = 'Actual') +
```

```
    theme_minimal() +
    scale_y_reverse()
}

plot1 <- plot_confusion_matrix(knnResult1, "KNN Result 1")
plot2 <- plot_confusion_matrix(knnResult2, "KNN Result 2")
plot3 <- plot_confusion_matrix(ldaResult1, "LDA Result 1")
plot4 <- plot_confusion_matrix(ldaResult2, "LDA Result 2")
plot5 <- plot_confusion_matrix(cartResult1, "CART Result 1")
plot6 <- plot_confusion_matrix(cartResult2, "CART Result 2")

plot1
plot2
plot3
plot4
plot5
plot6


# error rate comparison plot

error_rates <- data.frame(
  Model = c("KNN 1", "KNN 2", "LDA 1", "LDA 2", "CART 1", "CART 2"),
  Error = c(knntest.error1, knntest.error2, ldatest.error1, ldatest.error2, carttest.error1,
carttest.error2)
)

ggplot(error_rates, aes(x = Model, y = Error, fill = Model)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  labs(title = "Comparison of Test Errors", x = "Model", y = "Test Error") +
  theme_minimal()
```

References

https://www.geeksforgeeks.org/divide-a-vector-into-ranges-in-r-programming-cut-function/

https://www.rdocumentation.org/packages/knitr/versions/1.45/topics/kable

https://www.geeksforgeeks.org/joining-of-dataframes-in-r-programming/

https://scales.r-lib.org/

https://www.projectpro.io/data-science-in-r-programming-tutorial/ggvis

https://r-spatial.github.io/sf/

https://r-spatial.org/r/2018/10/25/ggplot2-sf.html

https://cran.r-project.org/web/packages/sf/index.html

https://www.geeksforgeeks.org/melting-and-casting-in-r-programming/

https://ggplot2.tidyverse.org/reference/scale_gradient.html

https://ggplot2.tidyverse.org/reference/labs.html