Insights into Accident Severity Prediction: A Comprehensive Review of Road Safety

Author: G.R.Devakumar 21BLC1185

Problem Statement:

Accidents on the roads are a significant concern that affects individuals, communities, and nations worldwide. The severity of these accidents can have a profound impact on the affected individuals, their families, and the overall society. Understanding the factors that contribute to accident severity is crucial for developing effective preventive measures and improving road safety. In this paper, we aim to predict accident severity and conduct a socioeconomic analysis to identify the underlying factors that influence the severity of accidents.

Objective:

The main objectives of this literature review are as follows:

- To predict the severity of road accidents with a high degree of accuracy using available data and advanced analytical techniques.
- To analyze gender and age of casualties to identify demographic patterns and trends in accidents.
- To identify the key contributing factors that influence accident severity, such as demographic characteristics, road infrastructure, vehicle attributes, and socioeconomic indicators.

Data Author:

Department for Transport - United Kingdom **Dataset**

Name:

Road Safety Data - Casualties 2022 - Provisional Mid Year

Description:

The dataset comprises comprehensive information on road accidents and related casualties that occurred in the UK during the year 2022.

It includes details such as accident index, accident year, accident reference, vehicle reference, casualty reference, casualty class, sex and age of the casualty, casualty severity, pedestrian information, vehicle type, and other relevant attributes.

The dataset provides a valuable resource for analyzing accident severity and understanding the underlying factors contributing to different levels of severity.

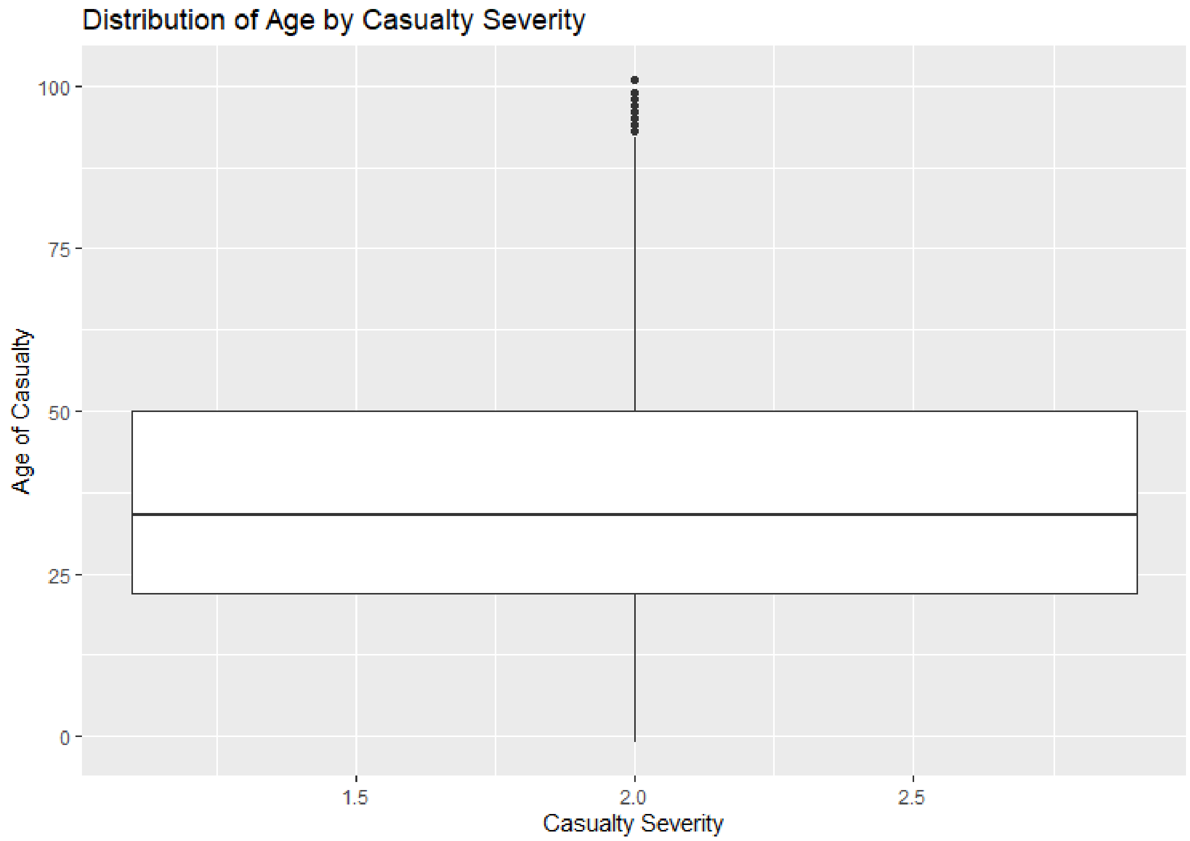
Program :

```r
# Load required packages
library(ggplot2)

# Load the dataset
dataset <- read.csv("C:/Users/Dev/Desktop/dft.csv")



# Dataset visualization - Box plot
ggplot(dataset, aes(x = casualty_severity, y = age_of_casualty)) +
  geom_boxplot() +
  labs(x = "Casualty Severity", y = "Age of Casualty") +
  ggtitle("Distribution of Age by Casualty Severity")
```



```
# Calculate the correlation coefficient

correlation_coefficient <- cor(dataset$age_of_casualty, dataset$casualty_severity)


# Print the correlation coefficient

cat("Correlation Coefficient:", correlation_coefficient)



# Calculate the correlation coefficient

correlation_coefficient <- cor(dataset$casualty_home_area_type, dataset$casualty_imd_decile)


# Print the correlation coefficient

cat("Correlation Coefficient:", correlation_coefficient)
```

```
> # Calculate the correlation coefficient
> correlation_coefficient <- cor(dataset$age_of_casualty, dataset$casualty_severity)
>
> # Print the correlation coefficient
> cat("Correlation Coefficient:", correlation_coefficient)
Correlation Coefficient: -0.09018458>
>
> # Calculate the correlation coefficient
> correlation_coefficient <- cor(dataset$casualty_home_area_type, dataset$casualty_imd_decile)
>
> # Print the correlation coefficient
> cat("Correlation Coefficient:", correlation_coefficient)
Correlation Coefficient: 0.5311537
```

```
# Set a random seed for reproducibility

set.seed(123)


# Split the dataset into training and test sets (70% for training)

train_indices <- sample(1:nrow(dataset), 0.7 * nrow(dataset))

train_data <- dataset[train_indices, ]

test_data <- dataset[-train_indices, ]


# Linear Regression


# Create a linear regression model

linear_model <- lm(casualty_severity ~ age_of_casualty + casualty_type, data = train_data)


# Make predictions on the test set

predicted_values_lr <- predict(linear_model, newdata = test_data)
```

```
# Calculate the mean squared error (MSE) for linear regression

mse_lr <- mean((predicted_values_lr - test_data$casualty_severity)^2)


# Display the MSE for linear regression

print(paste("Mean Squared Error (MSE) for Linear Regression:", mse_lr))
```

```
> # Linear Regression
>
> # Create a linear regression model
> linear_model <- lm(casualty_severity ~ age_of_casualty + casualty_type, data = train_data)
>
> # Make predictions on the test set
> predicted_values_lr <- predict(linear_model, newdata = test_data)
>
> # Calculate the mean squared error (MSE) for linear regression
> mse_lr <- mean((predicted_values_lr - test_data$casualty_severity)^2)
>
> # Display the MSE for linear regression
> print(paste("Mean Squared Error (MSE) for Linear Regression:", mse_lr))
[1] "Mean Squared Error (MSE) for Linear Regression: 0.198500158151005"
```

```
# Convert "sex_of_casualty" to numeric

train_data$sex_of_casualty <- as.numeric(as.character(train_data$sex_of_casualty))

unique(train_data$sex_of_casualty)



# Logistic Regression

# Perform logistic regression

logistic_model <- glm(sex_of_casualty ~ age_of_casualty + casualty_severity, data = train_data,
family = "binomial")


# Make predictions on the test set

predicted_values_logreg <- predict(logistic_model, newdata = test_data, type = "response")


# Convert predicted probabilities to predicted class labels

predicted_labels_logreg <- ifelse(predicted_values_logreg >= 0.5, 1, 0)


# Calculate accuracy for logistic regression

actual_labels <- test_data$sex_of_casualty

accuracy_logreg <- sum(predicted_labels_logreg == actual_labels) / length(actual_labels)
```

Display the accuracy for logistic regression

print(paste("Accuracy for Logistic Regression:", accuracy_logreg))

```
>
> # Display the accuracy for logistic regression
> print(paste("Accuracy for Logistic Regression:", accuracy_logreg))
[1] "Accuracy for Logistic Regression: 0.625719873954145"
```

Calculate the number of matches and mismatches

matches <- sum(predicted_subset == actual_subset)

mismatches <- sum(predicted_subset != actual_subset)

Create a data frame to store match and mismatch counts

count_df <- data.frame(Matches = matches, Mismatches = mismatches)

Create a bar plot to visualize the counts

barplot(t(count_df), beside = TRUE, col = c("green", "red"), names.arg = c("Count"),

ylim = c(0, max(matches, mismatches)), ylab = "Frequency",

main = "Match vs Mismatch (Sex of Casualty)")

legend("topright", legend = c("Matches", "Mismatches"), col = c("green", "red"), fill = c("green", "red"))