ASSESSMENT 2: DATA ANALYSIS

Part A: Proposal

36103 Statistical Thinking for Data Science

University Of Technology Sydney

Group Name

Retail Group

Group Members

Leah Nguyen

Tony Tan

Anmol Mittal

Ben McKinnon

Kasun Caldera

Manasa Burli Nagendra

Paul Touhill

Date

September 5, 2021

Contents

1 Introduction

Deaths from road traffic crashes continue to be major public health problem across Australia (institue of Health & Welfare 2021). Road traffic fatalities result from a complex interaction of human, technical and environmental factors, and different causes behind road vehicle crashes require different measures to reduce their impacts.

1.1 Stakeholders

The findings are relevant in the context of impact-based warnings for both road users, road maintenance and traffic management authorities, as well as emergency services.

1.2 Methodology

CRISP-DM methodology will be used for this project which is used in most of the data science projects in the industry (Wijaya 2021). The six steps are as follows:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation
- Deployment (Not in the scope of this assignment)

1.3 Business Understanding (Research question)

Our group study assesses how strongly the variation between some environmental factors and the increased risk of vehicle crash-related mortality in Victoria, Australia. Furthermore, we will use machine learning approaches to investigate the factors which cause a death in an accident this area.

2 Data Understanding

2.1 Datasets examined and their analysis

Various sources of data were examined and explored for the topic related to car accidents. We looked into car accidents historical data of Victoria, as well as NSW. We also looked into more peripheral data that are theorized to be related to car accidents, such as data of weather and speed camera presence. The following datasets were chosen for analysis.

2.1.1 Accident data from Victoria

Accident datasets are datasets with anonymized information from government of all road accidents in Victoria, Australia from 2006 until 2020 is used (Department of Transport 2006). The data provided allows users to analyse Victorian fatal and injury crash data based on time, location, conditions, crash type, road user type, object hit etc.

2.1.2 Weather

Weather datasets contains about daily weather observations describing daily mean, maximum, and minimum temperature and precipitation for the year of 2006 - 2020 from many locations across Australia. We collected the data using the Visual Crossing Weather API to perform a preliminary exploration to determine what, if any, predictive value weather has on crash and injury data when analyzed at the level of incident.

2.1.3 Analysis

The dataset will be explored, summarized, cleaned and visualized using various libraries in R. We will then prepare the final dataset for modelling.

3 Data Preparation

3.1 Accident data

The car accident data will be sourced directly from the Victorian Government data website. Not all files that are extracted from the website will be used in the analysis. After selecting specific files by their file name, they will be loaded into separate variables ready to be merged. What's considered to be the initial relevant variables are kept in the final data for analysis, though we may keep them in case for future analysis. And from here we can then attempt to find further interesting information from exploratory data analysis (EDA) techniques, such as to find correlation using correlation matrixes, scatterplots and contingency tables.

3.2 Weather data

The visual crossing weather API has a query which can search by location including the use of postcode. Using a collection of all the distinct postcodes from the node dataset, we can then make an API request for the address. As the API response contains a list of records for the date range specified in its query, we can then match each of our accident records with the weather using the postcode for location and the date. As the process of scraping data from the API can be long, the program has broken down the collection to collect the weather data for all the suburbs per year. Therefore, a process of merging the weather data of each year and splitting the columns of the address in order to match by postcode, and finally matching the postcode was done. This will also be part of the final data for EDA.

3.3 Addressing the rare class problem

Within our analysis, we may encounter "the rare class problem." 98% of the accidents in our data have no fatalities and only 2% resulted in a death. So, the model could predict an outcome mostly as no death even when it should end up in a death. If this problem is encountered, we will try to reduce our sample to make the data set more balanced, as well as using the lift method.

4 Model

4.1 Classification Model – Generalised Linear Model

To solve this problem a logistic regression model will be built. The years 2006 – 2016 will be used as a training set and years 2017-2018 will be used as the testing set, with 2020 saved for cross validation. A binary variable to identify if there was a fatality in the car accident will be created and set as our target variable. To test several model fits, we will use backward selection method to determine which features are contributing factors to causing or preventing fatalities and should be used in the final model. The model accuracy goal is set to be above 80%.

4.2 Evaluation

To evaluate each model fit the following evaluation methods will be used:

- **Confusion Matrix:** A display of the record counts by their predicted and actual classification (Chen 2021).
- **Accuracy:** (TruePositve + TrueNegative) / SampleSize (Chen 2021).
- **Sensitivity:** The percentage of all 1's that are correctly classified as 1's (Chen 2021).
- **Specificity:** The percentage of all 0's correctly classified as 0's (Chen 2021).
- **Precision:** The percent (proportion) of predicted 1s that are actually 1s (Chen 2021).
- **ROC Curve/AUC:** A plot of sensitivity versus specificity (Chen 2021).
- **Lift:** A measure of how effective the model is at identifying rare cases at different probability cutoffs (Chen 2021).

5 Conclusions and Limitations

The project is based on the rationale to reduce the number of deaths caused by motor vehicle accidents on Australian roads. The data has been collected from the Victoria government accidents data using an API. The data will be cleaned, explored and visualized using various R libraries and will be prepared for modelling. The major models used will be a linear regression models aiming to classify the accidents causing or not causing deaths and then finding the major factors leading to a death. The model will be evaluated using various techniques. The model efficiency goal is set to be above 80% accurate.

The major limitation of the project is that we do not have data where the accidents are not happening, so we had to choose the target variable as the fatality in an accident. This can cause a bias as we might have no accidents with the same conditions and factors. We had to assume that the probability of occurrence of an accident in the same conditions is 1.

6 Bibliography

Chen, L.-P. (2021), 'Practical statistics for data scientists: 50+ essential concepts using r and python', *Technometrics* **63**(2), 272–273.
URL: *https://doi.org/10.1080/00401706.2021.1904738*

Department of Transport (2006), 'Victoria Car Accident Dataset 2006 - 2020'.
URL:
https://vicroadsopendatastorehouse.vicroads.vic.gov.au/opendata/Road_Safety/ACCIDENT.zip

institue of Health & Welfare (2021), Injury in australia: transport injuries, Report, AIHW.
URL: *https://www.aihw.gov.au/reports/injury/transport-injuries*

Wijaya, C. Y. (2021), 'CRISP-DM Methodology For Your First Data Science Project'.
URL: *https://towardsdatascience.com/crisp-dm-methodology-for-your-first-data-science-project-769f35e0346c*

7 Appendix

```
# -----------------------------------
# Libraries
#------------------------------------
library(tidyverse)
library(foreign)
library(httr)
library(rjson)
library(here)
library(lubridate)
#-----------------------------------------
# Source Car Accident Data from Vicroads
#-----------------------------------------
# Set where zip file will be saved locally ----
file_path <- 'XXXXXXX'
setwd(file_path)
# Download and extract data  ----
url <-
'https://vicroadsopendatastorehouse.vicroads.vic.gov.au/opendata/Road_Safety/ACCIDENT.zip'
download.file(url, 'CarAccidentsData.zip')
unzip('CarAccidentsData.zip')
# Place selected files into variables  ----
f <- file.path(file_path, c("ACCIDENT.csv","ACCIDENT_LOCATION.csv",
                            'NODE.csv','PERSON.csv',"ATMOSPHERIC_COND.csv",
                            'ROAD_SURFACE_COND.csv','VEHICLE.csv'))
# Create names for the variables ----
names(f) <- gsub(".*/(.*)\\..*", "\\1", f)

# Read files into variables ready for analysis
for (i in 1:length(f)){
    x= read_csv(f[i])
    names(x)<- gsub(' ','_', names(x))
```

```r
    assign(names(f[i]),x)
    remove(x)
  }
#------------------------------------------------
# Source weather and postcode data from API
#------------------------------------------------
# Source Post Code Data ----
nodes <- read_csv(here("data/ACCIDENT/NODE.csv"))
postcodes<-unique(nodes$POSTCODE_NO)
postcodesJson <- toJSON(postcodes, indent = 0, method = "C")
write(postcodesJson, "data/ACCIDENT/postcodes.json")
# Load postcodes  ----
postcodes <- fromJSON(
  file = "data/ACCIDENT/postcodes.json") # json with list of postcodes
# API Key ---
RAPIDAPI_KEY = 'd1d5ff8ef9msh03f3fb1acd367a2p14e523jsnf314364cf14f'
# Create a function to be used to call API ----
request_by_postcode_and_year <- function(postcode, yearStart){
  base_url <- "https://visual-crossing-weather.p.rapidapi.com/"
  path <- "history"
  query_string <- list(
    startDateTime = sprintf('%s-01-01T00:00:00', yearStart),
    aggregateHours = '24',
    location = sprintf('%s,VIC,AUS', postcode),
    endDateTime = sprintf('%s-12-31T00:00:00', yearStart),
    contentType = 'csv',
    shortColumnNames = '0'
  )
# Send GET request to weather API ----
  response <- GET(
    url = base_url,
    path = path,
    add_headers(
      'x-rapidapi-host' = 'visual-crossing-weather.p.rapidapi.com',
      'x-rapidapi-key' = RAPIDAPI_KEY
    ),
    query = query_string,
    content_type('application/octet-stream'))

  text <- content(response, "text")
  return(text);
}
# Create function to combine API requests ----
gather_by_postcodes_and_year <- function(postcodes, year){
  final_df <- data.frame(
    matrix(ncol = 0, nrow = 0)) # initialize empty data frame
  for(postcode in postcodes){
    csv_response_text = request_by_postcode_and_year(postcode, year)
    df_from_response <- read_csv(csv_response_text)
    final_df <- bind_rows(final_df, df_from_response)
    cat("|") # some feedback on console
    Sys.sleep(1) # go easy on the api just in case
  }
  return(final_df)
}
# Create a function to create csv from df, data folder if it doesn't exist ----
create_csv <- function(df, file_name){
  if(!dir.exists("data")){
    dir.create("data")
  }
  write_csv(df, sprintf("data/%s", file_name))
}
# Create a function to extract data by postcode and year ----
create_csv_data_for_year <- function(year){
  combined_postcodes_df = gather_by_postcodes_and_year(postcodes, year)
  new_file_name = sprintf("sample_weather_%s.csv", year)
  create_csv(combined_postcodes_df, new_file_name)
}
# Extract weather API data by year and place into csv ----
if(!dir.exists("data/weather_data")){ # don't run to gather weather data
                                      # if folder already exists
 for(year in 2006:2020){
    create_csv_data_for_year(year)
 }
}
# Read consolidated API weather data place into one csv file ----
# setwd()
WEATHER_DATA <- list.files(pattern = '*.csv') %>%
                            map_df(~read_csv(.))
# Remove Spaces from heading names ----
names(WEATHER_DATA) <- gsub(' ', '', names(WEATHER_DATA))
# Convert columns with numerical data to correct data type ----
WEATHER_DATA[,c(1,5:20)] <- sapply(WEATHER_DATA[,c(1,5:20)], as.numeric)
# Select relevant columns by index number ----
WEATHER_DATA <-  WEATHER_DATA[,c(1,4,7:9,11,15,18,19,21,27)]
# Clean and convert date column ----
WEATHER_DATA$Datetime <- str_extract(WEATHER_DATA$Datetime, "\\d+/\\d+/\\d+")
WEATHER_DATA$Datetime <- as.Date(WEATHER_DATA$Datetime, format =  "%m/%d/%Y")
# Weather API data is ready to be added to Car Accident data ----
WEATHER_DATA
#----------------------------------------------
# Combined All Data To Begin Analysis
#----------------------------------------------
# Combine Car Accident Data ----
BASE <-  left_join(PERSON, ACCIDENT, by='ACCIDENT_NO') %>%
          left_join(x=., ROAD_SURFACE_COND, by='ACCIDENT_NO') %>%
           left_join(x=., ACCIDENT_LOCATION, by='ACCIDENT_NO')  %>%
            left_join(x=., NODE, by='ACCIDENT_NO') %>%
              left_join(x=.,ATMOSPHERIC_COND,by='ACCIDENT_NO') %>%
                left_join(x=., VEHICLE %>%
                                  select(ACCIDENT_NO,VEHICLE_ID,
                                         VEHICLE_YEAR_MANUF,
                                         Road_Surface_Type_Desc,
                                         VEHICLE_BODY_STYLE,
                                         VEHICLE_MAKE, VEHICLE_MODEL,
                                         NO_OF_CYLINDERS,
                                         TOTAL_NO_OCCUPANTS),
                          by= c('ACCIDENT_NO'='ACCIDENT_NO',
                                'VEHICLE_ID'='VEHICLE_ID' ))
# Clean and convert date column ----
BASE$ACCIDENTDATE <- str_extract(BASE$ACCIDENTDATE, "\\d+/\\d+/\\d+")
BASE$ACCIDENTDATE <- dmy(BASE$ACCIDENTDATE)
# Combine base data with API Weather data ----
data <- left_join(BASE, WEATHER_DATA, by= c('POSTCODE_NO'='Postcode',
                                            'ACCIDENTDATE'='Datetime'))
# Clean up unused variables ----
remove(PERSON, ACCIDENT, ROAD_SURFACE_COND, ACCIDENT_LOCATION,NODE,
       ATMOSPHERIC_COND,VEHICLE, WEATHER_DATA, BASE)
# Reorder columns ----
data <- data[,c(1,18,19,22,23,2,4:9,12,34:39,21,25,42:44,72,
                74:76,15,50:53,62,73,46,71,77,78,31,32,79:87)]
# Create Target Variable ----
data<- data %>%
        mutate(FATAL_ACCIDENT = if_else(NO_PERSONS_KILLED>0,"Y","N"),
              FATAL_ACCIDENT = factor(FATAL_ACCIDENT, levels = c("Y", "N"))) %>%
        relocate(FATAL_ACCIDENT, .after = ACCIDENT_NO )
# data is ready for EDA -----
data
```